VLSI RESIDUE MULTIPLIER MODULO A FERMAT NUMBER

I. S. Reed, T. K. Truong
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089

J. J. Chang, H. M. Shao, I. S. Hsu
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

ABSTRACT

Multiplication is central in the implementation of Fermat number transforms (FNT) and other residue number algorithms. There is need for a good multiplication algorithm which can be realized easily on a VLSI chip. In this paper, the Leibowitz multiplier [1] is modified to realize multiplication in the ring of integers modulo a Fermat number. The advantage of this new algorithm over Leibowitz's algorithm is that Leibowitz's algorithm takes modulo after the product of multiplication is obtained. Hence time is wasted. In this new algorithm, modulo is taken in every bit operation when performing multiplication. Therefore no time is wasted in this respect. Furthermore, this algorithm requires only a sequence of cyclic shifts and additions. The design for this new multiplier are regular, simple, expandable and therefore, suitable for VLSI implementation.

I. INTRODUCTION

Fermat number transforms (FNTs) were developed to compute cyclic convolutions [2-3] and to encode and decode a Reed-Solomon (R-S) codes in $GF(F_t)$ [4]. It was shown [5] that a 256- point Fermat transform in $GF(F_3)$, where $F_3 = 2^8 + 1 = 257$ is used to encode and decode a (256, 224) R-S code. Recently Roefs and Best showed in [6] that such R-S transform decoder over $GF(F_3)$ concatented with Viterbi decoder satisfies the operational requirements of the European Space Agency (ESA). The operations needed to perform this Fermat transform requires multiplication by power of 3 and addition in GF(256). Leibowitz [1] proposed the diminished -1 representation for binary arithmetic operations modulo F_t. Based on the ideas in [1], a new algorithm is developed to realize a multiplier over the ring of integers modulo a Fermat number. This algorithm requires only cyclic shifts and additions. An example, illustrating both the pipeline and systolic array aspects of this structure, is given for a multiplier in the field $GF(2^{2^2} + 1)$. Finally, a single working VLSI chip for a multiplier modulo a Fermat prime $F_2 = 2^4 + 1 = 17$ is given in this paper.

*This work was supported in part by NASA Contract No. NAS 7-100 and in part by Air Force Systems Command F19628-83-K-0009. A preliminary version of this paper was presented at 1985 IEEE International Conference on Acoustics, Speech and Signal Processing, March 26-29, Tampa, Florida.

II. MULTIPLICATION MODULO F_t

In this section, a new algorithm is developed for multiplication in the ring of integers modulo $F_t=2^{2^t}+1$. This new algorithm is illustrated by the example for t=2. The same structure clearly extends to more general multiply algorithms over F_t.

To perform efficiently the binary arithmetic operations modulo F_t, the diminished -1 representation proposed by Liebowitz [1] is used. Table 1 shows the correspondence of elements in $GF(2^4+1)$ with their decimal equivalents in a normal binary representation and with their values in the diminished -1 representation. In the Liebowitz diminished -1 representation the most significant bit (MSB) can be viewed as the zero-detection bit.

To realize this new algorithm for multiplication let A,B be two binary integers and A-1, B-1 their diminished -1 representations. It is now desired to perform a multiplication of the two positive numbers A-1 and B-1 in their diminished -1 representation. Note that if the MSB of either A-1 or B-1 is one, then the multiplication is inhibited, and the product is zero.

To accomplish this new multiplication process one first translates B-1 from the diminished -1 representation to the normal binary representation, B. Then the multiplication of A-1 and B is performed as follows:

$$(A-1)\cdot B=(A-1)\left(\sum_{k=0}^{4} b_k 2^k\right)=\sum_{k=0}^{4} b_k 2^k\cdot(A-1) \qquad (1)$$

Multiply the diminished -1 numbers by the different powers of two in Eq. (1) so that the result is a diminished -1 number. This is achieved by the

identity $2^k(A-1)+(2^k-1)=2^kA-1$ for $0\leq k\leq 4$. Thus when considered as a sum of diminished numbers, Eq. (1) becomes

$$(A-1)B=\sum_{k=0}^{4} b_k\cdot(2^kA-1)=\sum_{k=0}^{4}(d_k-1) \qquad (2)$$

where $d_k-1=b_k\cdot(2^kA-1)$ for $0\leq k\leq 4$. For d_k-1 for $0\leq k\leq 4$ to be diminished -1 numbers they must add correctly pairwise in accordance with the formula,

$$(d_i-1)+(d_j-1)+1=(d_i+d_j-1) \qquad (3)$$

It is readily verified that this holds in all four cases of (d_i, d_j). Hence diminished -1 addition in (2) can be performed recursively in the following manner:

$$(A-1)\cdot B=(((((d_0-1)+(d_1-1)+1)+(d_2-1)+1)+ (d_3-1)+1)+(d_4-1)+1) \qquad (4)$$

where $d_k-1=b_k(2^kA-1)$. From (5) the following identity is obtained for the product A•B diminished by 1;

$$(A-1)\cdot B=b_0\cdot(A-1)+b_1\cdot(2A-1)+b_2\cdot(2^2A-1)+ b_3\cdot(2^3A-1)+b_4\cdot(2^4A-1)+4$$
$$=A\cdot B-S+4 = (A\cdot B-1)-S+5 \qquad (5)$$

where $S=b_0+b_1+b_2+b_3+b_4$. It follows from (5) that A•B in diminished -1 notation is

$$(A\cdot B-1)=(A-1)\cdot B-D-2+1 \qquad (6)$$

where $D=4-S$ such that $0\leq D\leq 4$. Since $2^4+1\equiv 0 \bmod F_2$, one has $-D-2=2^4+1-D-2=(2^4-D-1)=\overline{D}$, the binary one's complement of D. Hence, by (6), the diminished -1 representation of A•B becomes

$$C=(A\cdot B-1)+(A-1)\cdot B+\overline{D}+1 \qquad (7)$$

A substitution of Eq. (5) into Eq. (7) yields

$$(A\cdot B-1)=(((((\overline{D}+b_0\cdot(2^0A-1)+1)+b_1\cdot(2^1A-1)+1)+ b_2\cdot(2^2A-1)+1)+b_3\cdot(2^3A-1)+1)+b_4\cdot(2^4A-1)+1) \qquad (8)$$

as the new multiply algorithm.

Let $C_0=\overline{D}$. Then the multiplication algorithm in Eq. (8) can be put into the following recursive form

$$C_{k+1}=C_k+b_k\cdot(2^{k+1}\cdot A-1)+1 \quad \text{for} \quad 0\leq k\leq 4 \qquad (9a)$$

If one successively computes C_{k+1} in (9a) for $0\leq k\leq 4$, then the required result is obtained as follows:

$$C_5=C_4+b_4(2^4A-1)+1=(A\cdot B-1)=C \qquad (9b)$$

Example 1 (A recursive diminished -1 multiply algorithms):

Let A-1 = 0 1 0 1 0, B-1 = 0 0 1 0 1, Compute

C=(A•B-1)=0 1 0 1 0 × 0 0 1 0 1 modulo 2^4+1.

To compute C, one first translate B-1 to B. That is, B=B-1+1=(0 0 1 0 1)+1=0 0 1 1 0=b_4 b_3 b_2 b_1 b_0. From (8), the sequence of computations for 0 1 0 1 0 × 0 0 1 0 1 is then as follows:

0 1 1 0 1	$C_0=\overline{D}$=0 1 1 0 1
+ 0 0 0 0 0	$b_0\cdot(2^0A-1)$=0(0 1 0 1 0)=
0 1 1 0 1	0 0 0 0 0
+ 1	
0 1 1 1 0	$C_1=C_0+b_0\cdot(2^0A-1)+1$
+ 0 0 1 0 0	$b_1\cdot(2^1A-1)$=1•(0 0 1 0 0)=
1 0 0 1 0	0 0 1 0 0
+ 0	
0 0 0 1 0	$C_2=C_1+b_1\cdot(2^1A-1)+1$
+ 0 1 0 0 1	$b_2\cdot(2^2A-1)$=1•(0 1 0 0 1)=
0 1 0 1 1	0 1 0 0 1
+ 1	
0 1 1 0 0	$C_3=C_2+b_2\cdot(2^2A-1)+1$
+ 0 0 0 0 0	$b_3\cdot(2^3A-1)$=0•(0 0 0 1 0)=
0 1 1 0 0	0 0 0 0 0
+ 1	
0 1 1 0 1	$C_4=C_3+b_3\cdot(2^3A-1)+1$
+ 0 0 0 0 0	$b_4\cdot(2^4A-1)$=0•(0 0 1 0 1)=
0 1 1 0 1	0 0 0 0 0
1	
0 1 1 1 0	$C_5=C_4+b_4(2^4A-1)+1=C$

Thus C=0 1 1 1 0 is the desired result of 0 1 0 1 0 times 0 0 1 0 1, modulo 2^4+1 in diminished -1 notation.

III. A VLSI STRUCTURE FOR IMPLEMENTING MULTIPLICATION MODULE F_t

Example 1 of the new diminished -1 multiply algorithm in the previous section shows that diminished -1 additions require the addition of the complement of an end around carry to its sum. A considerable speed improvement can be obtained by performing this operation simultaneously with the summation. A modified algorithm with this simultaneous addition is given for the previous example as follows:

Example 2 (Modified recursive diminished -1 multiplication):

1 1 1 0 1	C_0=1 0 0 0 0+$\overline{D}$=1 1 1 0 1
0 0 0 0 0	$b_0\cdot(2^0A-1)$=0(0 1 0 1 0)=
+ 0	0 0 0 0 0
0 1 1 0 1	$C_1=C_0+b_0\cdot(2^0A-1)+1$
0 0 1 0 0	$b_1(2^1A-1)$=1·(0 0 1 0 0)=
+ 1	0 0 1 0 0
1 0 0 1 0	$C_2=C_1+b_1(2^1A-1)+1$
0 1 0 0 1	$b_2(2^2A-1)$=1·(0 1 0 0 1)=
+ 0	0 1 0 0 1
0 1 0 1 1	$C_3=C_2+b_2(2^2A-1)+1$
0 0 0 0 0	$b_3(2^3A-1)$=0·(0 0 0 1 0)=
+ 1	0 0 0 0 0
0 1 1 0 0	$C_4=C_3+b_3(2^3A-1)+1$
0 0 0 0 0	$b_4(2^4A-1)$=0·(0 0 1 0 1)=
+ 1	0 0 0 0 0
0 1 1 0 1	$C_5=C_4+b_4(2^4A-1)+1$
0 0 0 0 0	$0\cdot(2^5A-1)$=0·(0 1 0 1 1)=
+ 1	0 0 0 0 0
0 1 1 1 0	$C=C_5+1$.

A possible VLSI structure for Example 2 is presented in Figure 1.

In Figure 1, A, B and C are 4-bit, 6-bit, and 5-bit registers, respectively. Initially registers A, B and C contains the multiplicand in the diminished -1 representation, the multiplier in normal representation, and $2^4+\overline{D}$, respectively. At the very same moment $C_{k+1}=C_k+b_k(2^kA-1)$ is computed and loaded into the C register. Simultaneously the diminished -1 multiplication of (A-1) by 2 is performed first by a left cyclic shift of the 4 least significant bits of the register A with the A_3-bit circulated into the first significant bit complemented. Also at the same time register B is shifted right by one bit. These operations are continued repetitively until the MSB of the register B is shifted out The desired final result 0 1 1 1 0 after 5 iterations is obtained in register C.

The layout of the structure in Fig. 1 has been completed with the use of the CAESER design tool [7]. The final layout of the multiplication chip is shown in Fig. 2. Both the logic and circuit level simulations are performed using logic simulator "esim" [8] and circuit simulator "spice" [9]. The timing analysis was done with timing simulator "crystal" [10] as well. The working chip of a multiplication circuit for the Fermat number $F_2=2^4+1$ is shown in Fig. 3. The operating frequency is around 6 M HZ with 4 μm NMOS technology. The total number of transistors in this chip is about 630. The area of the chip is estimated to be about 40 mil×36 mil.

REFERENCES

[1] L. M. Leibowitz, "A Simplified Binary Arithmetic for the Fermat Number Transform" IEEE Trans, Acoustic, Speech and Signal Processing, Vol. ASSP-24, No. 5, pp. 356-359, Oct. 1976.

[2] C. M. Rader, "Discrete Convolutions via Mersenne Transforms" IEEE Trans. Computers, Vol. C-21, No. 12, pp. 1269-1273, Dec. 1972.

[3] R. C. Agarwal and C. S. Burns, "Fast Convolution Using Fermat Number Transforms with Applications to Digital Filtering" IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-22, No. 2, pp. 87-97, April 1974.

[4] I. S. Reed, T. K. Truong, and L. R. Welch, "The Fast Decoding of Reed-Solomon Code Using Fermat Number Transforms," IEEE Trans. Information Theory, Vol. IT-24, No. 4, pp. 497-499, July 1978.

[5] K. Y. Liu, I. S. Reed, and T. K. Truong "High-Radix Transforms for Reed-Solomon Codes Over Fermat Primes," IEEE Trans. on Information Theory, Vol. IT-23, No. 6, pp. 776-778, November.

[6] H. F. A. Roefs and A. R. Best, "Concatenated Coding on a Spacecraft to Ground Telemetry Channel Performance," Proc. ICC81, Denver, 1981.

[7] J. Ousterhout, "Editing VLSI Circuits with Caesar," Computer Science Division, Electrical Engineering and Computer Sciences, University of California, Berkeley, April 21, 1982.

[8] C. Tevman, "Esim-event Driven Switch Level Simulator," Dept. of Electrical Engineering, MIT, 1977.

[9] L.W. Negal and D.O. Pederson, "SPICE-Simulation Program with Integrated Circuit Emphasis," Memorandum No. ERL-M382, Electronics Research Laboratory.

[10] J. Ousterhout, "Using Crystal for Timing Analysis," Computer Science Division, Electrical Engineering and Computer Sciences, University of California, Berkeley, March 1983.

Table 1. The correspondence among decimal numbers, their values in the normal binary representation, and in the diminished -1 representation.

Decimal Number	Normal Binary Representation	Diminished -1 Representation
0	0 0 0 0 0	1
1	0 0 0 0 1	2
2	0 0 0 1 0	3
3	0 0 0 1 1	4
4	0 0 1 0 0	5
5	0 0 1 0 1	6
6	0 0 1 1 0	7
7	0 0 1 1 1	8
8	0 1 0 0 0	9 (-8)
9 (-8)	0 1 0 0 1	10 (-7)
10 (-7)	0 1 0 1 0	11 (-6)
11 (-6)	0 1 0 1 1	12 (-5)
12 (-5)	0 1 1 0 0	13 (-4)
13 (-4)	0 1 1 0 1	14 (-3)
14 (-3)	0 1 1 1 0	15 (-2)
15 (-2)	0 1 1 1 1	16 (-1)
16 (-1)	1 0 0 0 0	0

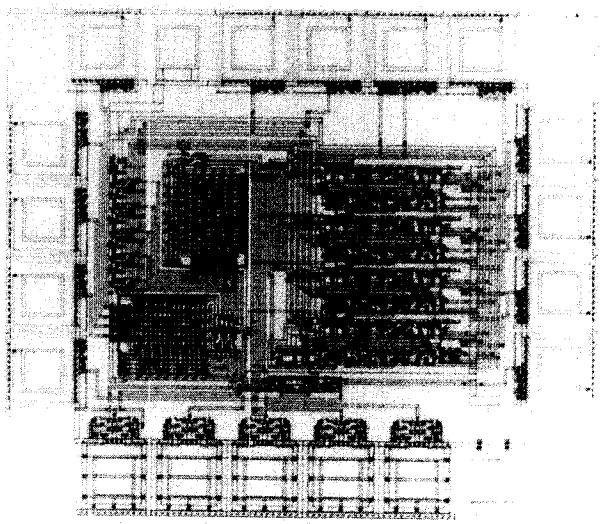

Fig. 2. VLSI layout of a multiplication circuit for the Fermat number $F_2=2^4+1$.

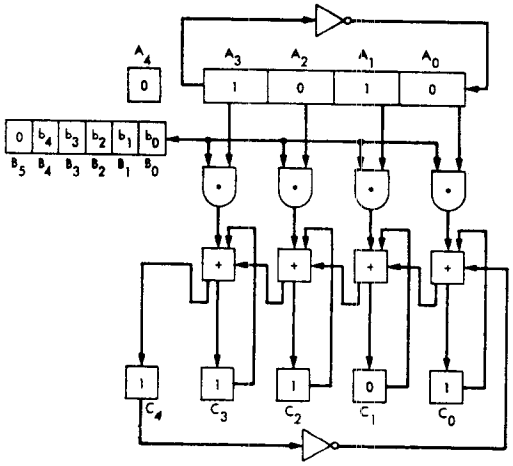

Fig. 1. The pipeline architecture for the implementation of multiplication modulo the Fermat number 2^4+1 using diminished -1 number representation.

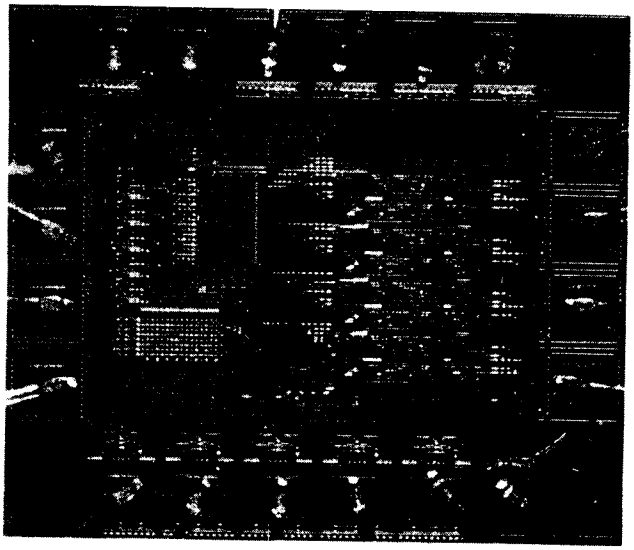

Fig. 3. Working VLSI chip of a multiplication circuit for the Fermat number $F_2=2^4+1$.