Design and Implementation of Low-Complexity Redundant Multiplier Architecture for Finite Field

Veerraju kaki
Electronics and Communication Engineering, India

Abstract- In the present work, a low-complexity Digit-Serial/parallel Multiplier over Finite Field is proposed. It is employed in applications like cryptography for data encryption and decryption to deal with discrete mathematical and arithmetic structures. The proposed multiplier utilizes a redundant representation because of their free squaring and modular reduction. The proposed 10-bit multiplier is simulated and synthesized using Xilinx Verilog HDL. It is evident from the simulation results that the multiplier has significantly low area and power when compared to the previous structures using the same representation.

Index Terms- Digit-Serial, Finite Field multiplication, Redundant Basis.

I. INTRODUCTION

In cryptography and coding theory, there are many applications using arithmetic operations for Finite Field [1]. In general, multiplication is extremely expensive in terms of time delay and physical area. Therefore, more focus is concentrated on designing high speed multipliers and on reduction of area [2]. The complexity mainly depends on representation of field elements. The most commonly used basis includes polynomial (PB), normal (NB), dual (DB) and redundant (RB) [3]. Dual and normal basis multipliers require a conversion of basis, in which heavily rely on the simplified polynomial. There is no need of conversion of basis in case of standard basis multipliers, the most commonly used multiplier is the polynomial basis multiplier due to their simple design and which also provide scalability.

Redundant basis (RB) is attractive when performing exponentiations and squaring operations [5]. The major advantage of redundant basis is squaring operation, as normal basis and also involves lower computational complexity. The multipliers of finite field are designed and classified into full parallel multipliers and word level multipliers [6]. The hardware used and power required by the bit-serial multipliers is less but it is slow.

To understand the complication between area and speed, the Digit-Serial multipliers are reported previously. These are scalable multipliers and classified into different forms .An effective multipliers which utilizes RB is presented previously. Multipliers with systolic structures are presented in [7].A comb architectures are also presented formerly. Word level multipliers over finite field with high speed are also reported .And several other multipliers are also been developed for reducing complexity.

In this paper, a low-complexity digit-serial/parallel is presented by utilizing a redundant basis over finite field (GF (2m)).The recursive decomposition scheme for digit-serial/parallel multipliers is same as the previous, where the multiplier is modified. In his work, a low-complexity multiplier is introduced which involves significantly low area and power complexities when verified with the previous techniques.

Organization of the paper is as follows: Review of existing digit-serial RB multiplier is presented in section 2. Proposed digit-serial RB multiplier mentioned in section 3. Implementation and Comparison are shown in section 4. The paper ends with conclusion in section 5.

II. EXISTING DIGIT SERIAL RB MULTIPLIER

In Digit serial RB multiplier [4] the input operands A and B are divided into the number of integers to attain Digit Serial Multiplication, to achieve the final product the partial products are summed.

Assume x is an n th root of unity, components in Finite Field GF (2m) are often described within the form:

$A=a_0+a_1x+a_2x^2+\cdots+a_{n-1}x^{n-1}$ (1)

Which a_i belongs to GF (2), for $0 \le i \le$ n - 1, alike the set $\{1, x, x^2, \cdots, x^{n-1}\}$ is outlined as the RB for Finite Field components, wherever n could be a positive number not below m. And just then $(m + 1)$ is prime and number 2 is a primitive root modulo $(m + 1)$, for a finite field, there being a type I Optimal Normal Basis (ONB) [8]. X is component of GF (2^m), & n= m+1.

Let A, B belongs to GF (2m) can be demonstrated in the form of RB:

$$A = \sum_{i=0}^{n-1} a_i x^i \quad (2)$$

$$B = \sum_{i=0}^{n-1} b_i x^i \quad (3)$$

Thus ai ,bi belongs to GF(2). , Let A and B are input operands which obtain product C, is demonstrated as follows

$$C = \mathrm{A.B} = \sum_{i=0}^{n-1}\left(x^i b_i\right).A \quad (4)$$

$$= \sum_{i=0}^{n-1}\left(\sum_{j=0}^{n-1} b_i x^{(i+j)}\right)a_j \quad (5)$$

$$= \sum_{j=0}^{n-1}\left(\sum_{i=0}^{n-1} b_{(i-j)_n} x^i\right)a_j \quad (6)$$

$$= \sum_{i=0}^{n-1}\left(\sum_{j=0}^{n-1} b_{(i-j)_n} a_j\right)x^i \quad (7)$$

Where $(i-j)_n$ denotes modulo n reduction. Define $C = \sum_{i=0}^{n-1} c_i x^i$, where $c_i \in$ GF (2), we have [10]. $c_i = \sum_{i=0}^{n-1} b_{(i-j)_n}\ a_j$ (8)

III. PROPOSED DIGIT-SERIAL RB MULTIPLIER

In Digit serial RB multiplier [4], to attain Digit Serial Multiplication both the inputs are divided into a number of units and the partial products related to these units are summed to achieve the desired product. Considering equations (1) and (7) of Digit serial RB Multiplier. Where $(i-j)_n$ denotes modulo n reduction. Define C in the form of:

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 & b_{n-1} & \cdots & b_1 \\ b_1 & b_0 & \cdots & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} \quad (9)$$

From (9), shifted form of the inputs bits B can be defined as follows

$$B^0 = \sum_{i=0}^{n-1} b_i^0 x^i = b_0 + b_1 x + \cdots + b_{n-1}x^{n-1} \quad (10)$$

$$B^1 = \sum_{i=0}^{n-1} b_i^1 x^i = b_{n-1} + b_0 x + \cdots + b_{n-2}x^{n-1} \quad (11)$$

……….

$$B^{n-1} = \sum_{i=0}^{n-1} b_i^{n-1} x^i = b_1 + b_2 x + \cdots + b_0 x^{n-1} \quad (12)$$

Where, $b_0^{i+1} = b_{n-1}^i$

$b_j^{i+1} = b_{j-1}^i$, for $1 \le j \le n-2$. (13)

The recursions on (13) can be extended further to have

$$b_j^{i+s} = \begin{cases} b_{n-s+j}^i, & for\ 0 \le j \le n-2 \\ b_{j-s}^i & other\ wise \end{cases} \quad (14)$$

Where 1 , Let P and Q are two integers alike n = QP + r, where 0 .for ease, assume that r = 0 and divide the input of A into Q units of vectors operands Au, where u= 0,1….Q-1follows:

$A_0 = [a_0\, a_Q \cdots a_{n-Q}]$ (15)

$A_1 = [a_1 a_{Q+1} \cdots a_{n-Q+1}]$ (16)

… … …

$A_{Q-1} = [a_{Q-1} a_{2Q-1} \cdots a_{n-1}]$ (17)

Identically, we produce the Q units of shifted vector operands Bu , where u= 0, 1, ,Q -1, follows:

$B_0 = [\, B^0 B^Q \cdots B^{n-Q}\,]$ (18)

$B_1 = [B^1 B^{Q+1} \cdots B^{n-Q+1}]$ (19)

… … …

$$B_{Q-1} = [B^{Q-1}\, B^{2Q-1} \cdots B^{n-1}]. \qquad (20)$$

The product C =AB which is obtained from (9) are broken down into products Q of vectors Au and Bu , where u = 0, 1, , Q-1 as:

$$C = AB = B_0 A_0^T + B_1 A_1^T + \cdots + B_{Q-1} A_{Q-1}^T$$

$$= \sum_{u=1}^{Q-1} B_u A_U^T = \sum_{u=0}^{Q-1} \overline{C}_u \qquad (21)$$

Where denotes $\overline{C_u} = B_u A_U^T$ (22)

Note that Au for u = 0, 1, , Q -1 is a P point bit – vector. Bu for u = 0, 1, , Q – 1 is a P bit-shifted forms of operand B.

The proposed structure shown in below Fig.1 is derived from the processor space flow graph in [4], consists of S nodes, M nodes and A nodes which S node performs shifting operation and M node, A node performs multiplication and addition operation.

The proposed Digit- serial RB multiplier consists of three block, bit-permutation block, partial product generation block and finite field accumulator block. The BPB performs the rewriting of inputs of B to consume the output according the shifting S node. The PPGM consists of AND cell which performs the multiplication operation and XOR cell which performs addition operation. And finite field accumulator block consistent with n-bit parallel accumulation units. The recent input which is received is added with past accumulated result, and the sum is retain in the register cell and used in the next cycle. And successive output is obtained. Fig.2 shows the structure of finite field accumulator which consists of XOR cell and register cells with n parallel input bits and n parallel outputs bits. In Fig.1 AND cell performs the multiplication of A input bits with the B input bits by bit-shifting form, XOR cell performs an addition operation of the outputs obtained from the AND cells, the operation can be done concurrently and the partial products obtained at the XOR cell.

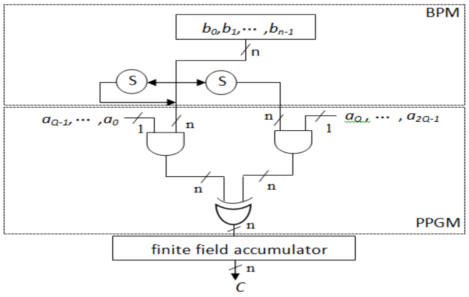


Figure 1. Proposed Digit-serial RB multiplier

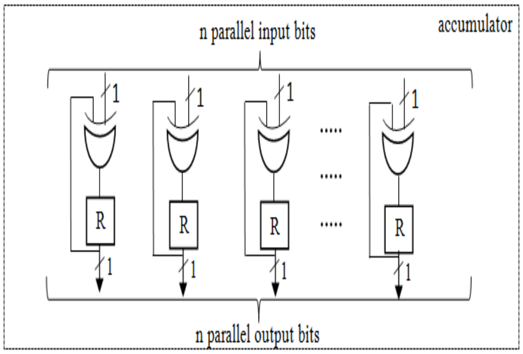
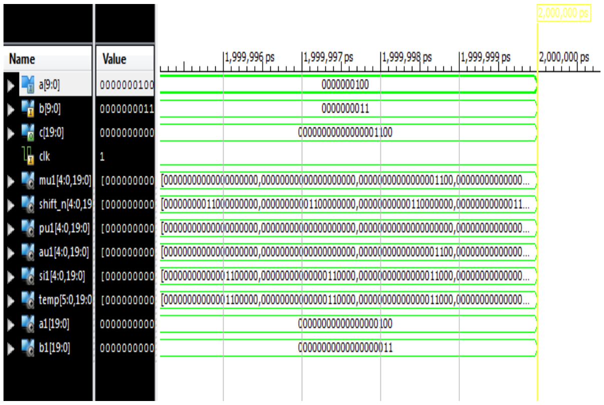


Figure 2. Structure of finite field accumulator

The partial products generated are fed to the finite field accumulator and result is accumulated and stored in register cell of the finite field accumulator and then finally the desired output is obtained. The partial products generated in this multiplier are lesser in numbers than those previous multipliers, and also reduces the area complexities and reduce in power.

IV. IMPLEMENTATION AND COMPARISON

The proposed digit-serial RB multiplier for 10-bit is coded using Verilog HDL in Xilinx ISE 12.2. The simulated results for the proposed structure are shown in Fig.3. In the above waveform the inputs are a and b and the output is c. when the 10-bit inputs a=0000000100 and b=0000000011 are given, by performing the shifting operation of the input operand b and multiplying with the each bit of the operand a, then each value is accumulated and stored in accumulator to obtain the desired output c=00000000000000001100. And the remaining values shown are the signals.



```
                                                                                                                      2,000,000 ps
Name            Value         |1,999,996 ps        |1,999,997 ps        |1,999,998 ps        |1,999,999 ps        2,000,000 ps
a[9:0]          0000000100                         0000000100
b[9:0]          0000000011                         0000000011
c[19:0]         0000000000                         00000000000000001100
clk             1
mu1[4:0,19:0]   [000000000   [00000000000000000000,00000000000000000000,00000000000000001100,0000000000000...
shift_n[4:0,19  [000000000   [00000000110000000000,00000000001100000000,00000000000110000000,0000000000011...
pu1[4:0,19:0]   [000000000   [00000000000000000000,00000000000000000000,00000000000000000000,0000000000000...
au1[4:0,19:0]   [000000000   [00000000000000000000,00000000000000000000,00000000000000001100,0000000000000...
si1[4:0,19:0]   [000000000   [00000000000001100000,00000000000000110000,00000000000000011000,0000000000000...
temp[5:0,19:0   [000000000   [00000000000001100000,00000000000000110000,00000000000000011000,0000000000000...
a1[19:0]        0000000000                         00000000000000000100
b1[19:0]        0000000000                         00000000000000000011
```

Figure 3. Simulation waveform of 10-Bit Proposed Digit-Serial RB multiplier

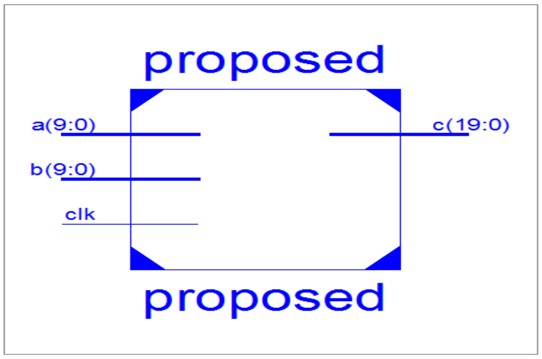


Figure 4. RTL Schematic of 10-Bit Proposed Digit Serial RB multiplier

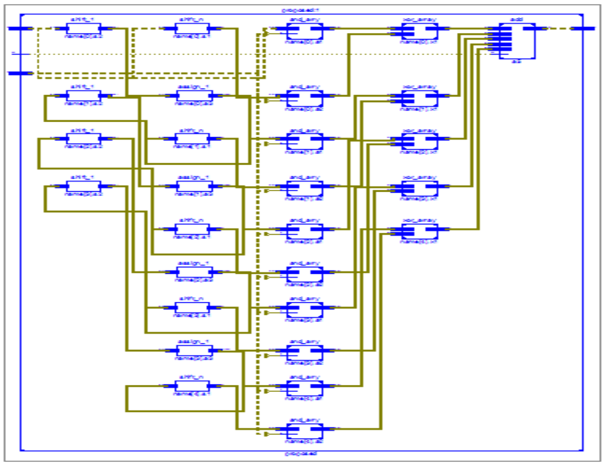


Figure 5. Detailed RTL Schematic of 10-Bit Proposed Digit-Serial RB multiplier

The RTL schematic of 10-bit Digit-Serial RB multiplier is shown in Fig.4, contain inputs a, b and clock and output c. Here a and b are the 10-bit inputs and which obtain the 20-bit output. The detailed view of 10-bit proposed digit-serial RB multiplier is in Fig.5, which gives the clear explanation of logic required.

Structures	Area(Number of slices)	Power(W)
Structure-I [4]	75	0.066
Structure-II [4]	105	0.067

Table 1.Comparison table

The 10-bit Structure-I, 10-bit structure-II [4] and 10-bit proposed digit-serial multiplier is implemented in Xilinx ISE .A table is formulated to show the results. The number of slices used in each structure is estimated and tabulated in table 1.This comparison table indicates the reduction in area .In similar manner; the power is also estimated and compared. The comparison result for number of slices that is area and power are also shown in fig.6 as a graphical plot for better comparison.

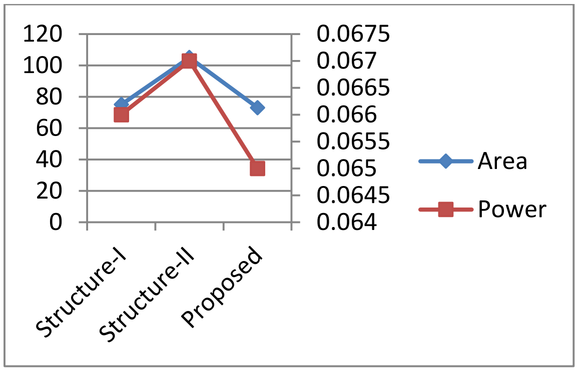


Figure 6. Area and Power comparison

V. CONCLUSION

The proposed 10-bit digit-serial RB multiplier is implemented. It is evident from comparison table that the performance of the proposed architecture is good with respect to speed and the area. The proposed 10-bit digit-serial multiplier using redundant basis is used based on application requirement and mostly in modern cryptographic applications. The proposed multiplier is derived to obtain less complexity than the previous multipliers

REFERENCES

[1] Swamy.M.N, May (2007) “Cryptographic applications of bhaskara equations” IEEE Trans.Circ.Sys.I, vol.54, no.7, pp. 927-928.

[2] M.Nikooghadam, March (2013) “ Low Power and High-Speed Design of a Versatile Bit-Serial Multiplier in Finite Field, the VLSI journal, vol 46, Issue 2, pp.211-217.

[3] L.S.Hsu, H.M.Shao, (1987) “Comparison of VLSI Architecture of Finite Field Multipliers Using Dual, Normal or Standard basis,” IEEE Tran. Comp, pp.63-75.

[4] Zhi-Hong Mao, J.Xie, Jan (2015) “High-throughput finite field multipliers using redundant basis for FPGA and ASIC implementation,” IEEE Trans. Cirt. Sys-I, vol.62, no.1, pp.110-119.

[5] M.Uma.Maheswari, S.Bhaskar, November (2014) “High Speed Finite Field Multiplier GF(2m) for Cryptographic Applications,” IJARECE, vol.3, Issue 11, pp.1705-1708.

[6] B.Sargunam, Dr.R.Dhanasekaran, April (2014) “Word Level Finite Field Multipliers Using Normal Basis,” JTAIT, vol.62, no.3, pp.805-811.

[7] J-S.Pan, Mehar. P.K, December (2013) “Low latency digit serial digit parallel systolic multipliers for large binary extension fields,” IEEE Trans. Circt.and Syst-I, pp.1-11.

[8] I.-C.Jou , C.-Y Lee, Sep. (2005) “Bit-Parallel Systolic Montgomery Multipliers for Special Classes of GF(2m) ,” IEEE Trans. Compt., vol.54,no.9, pp.1061-1070.