A Serial-In Parallel-Out Multiplier Using Redundant Representation For A Class of Finite Fields

Ashkan Hosseinzadeh Namin, Huapeng Wu, and Majid Ahmadi
Department of Electrical and Computer Engineering
University of Windsor
Windsor, Ontario N9B 3P4 Canada
Email: {hosse11,hwu,ahmadi}@uwindsor.ca

Abstract—A new Serial-In Parallel-Out finite field multiplier using redundant basis for a class of fields is proposed. It has been shown that the proposed architecture has higher speed in comparison to the previously proposed architecture using the same basis. Hardware realizations of the proposed multiplier and previously proposed multiplier along with their comparison is also presented.

I. INTRODUCTION

Finite field arithmetic plays an important role in public key cryptography, since two out of three main public key cryptosystems, El-Gamal and Elliptic curve, are based on finite field arithmetic [1], [2]. Research in this area is mostly concentrated around finding efficient methods to implement the arithmetic operations. The main arithmetic operation in finite field is multiplication since addition is done rather easily and other arithmetic operations, inversion and exponentiation, can be done with consecutive multiplications.

Finite field arithmetic uses the basis concept to represent field elements as it is used by normal arithmetic. Few different bases have been proposed which the most common ones are polynomial basis (or standard basis), normal basis, and dual basis. Note the basis used to represent the field elements has great impact on the finite field arithmetic operation performance [3].

Here we are mainly interested in a new proposed basis called redundant basis [4]. The main idea behind the redundant basis is to embed field elements in a larger ring to perform arithmetic operations. The ring used here is known as a cyclotomic ring that has a simple basis whose elements form a cyclic group. This idea was first proposed by Gao et al. to perform fast multiplication in normal basis [5], [6]. The architectures proposed for arithmetic operations in the cyclotomic ring are highly regular and suitable for hardware implementations.

Note that embedding a field in the cyclotomic ring is not a one to one mapping and the representation for each field element in the ring is not unique. This makes the representation to have certain amount of redundancy and hence called redundant basis representation. Redundant basis has the advantages that it not only offers free squaring operation as normal basis does, but also accommodates ring type operations.

Architectures proposed for hardware implementation of redundant basis multipliers can be categorized to three main categories, bit-level , word-level and full parallel. In this work we are interested in the bit-level category. Multipliers in the bit level category will receive the input coefficients bit by bit or will generate the product coefficients bit by bit. Consequently an n bit multiplier with take n clock cycles to finish one multiplication operation. These multipliers are considered to take small area of silicon chip and to be low power. Their main disadvantage is their low multiplication speed for large size multipliers.

Two types of bit-level architectures were proposed for hardware realization of redundant basis multipliers. The first one is parallel-in serial-out (PISO) and the second one is serial-in parallel-out (SIPO) architecture. In a parallel-in serial-out multiplier each bit of the output product is generated in one clock cycle. On the other hand in a serial-in parallel-out architecture, bits of the output product are incremented through clock cycles and become ready all at the same time.

In this paper a new serial-in parallel-out finite field multiplier using redundant basis for a class of fields is proposed. Comparisons with previously proposed architecture show that the proposed design is faster than the previous proposal. Hardware realization of the proposed multiplier is also presented and compared with hardware realization of the previously proposed architecture.

The organization of this article is as follows: In Section II, a brief review of redundant basis representation and multiplication is presented. A new serial-in parallel-out multiplication for a class of fields in redundant basis is proposed in Section III. The new architecture to preform serial-in parallel-out multiplication is presented in section IV. The architectural complexity of the proposed multiplier is compared with the previously proposed architecture and presented in this Section too. Hardware implementation for the proposed multiplier and the previous proposal are presented in Section V. Finally, a few concluding remarks are given in Section VI.

II. A BRIEF REVIEW OF REDUNDANT BASIS AND ITS ARITHMETIC IN $\mathbb{F}_{2^m}$

A. Redundant Basis for $\mathbb{F}_{2^m}$

Let $K$ be a field and $f(x) \in K[x]$ be a polynomial defined over $K$. Then the field that contains all the roots of $f(x)$ is
called the splitting field of the polynomial $f(x)$. The splitting field of $x^n - 1$ is called the $n$th cyclotomic field, denoted by $K^{(n)}$. Let $\beta$ be a primitive $n$th root of unity. Then $K^{(n)}$ is generated by $\beta$ over $K$ and elements in $K$ can be represented in the form

$$A = a_0 + a_1 \beta + a_2 \beta^2 + \cdots + a_{n-1} \beta^{n-1}, \ a_i \in K.$$  

Thus the set $[1, \beta, \beta^2, \ldots, \beta^{n-1}]$ can be viewed as a basis for $K^{(n)}$ [5], [6]. Since $1 + \beta + \beta^2 + \cdots + \beta^{n-1} = 0$ the representation of $A$ is not unique, for example, the two $n$-tuples $(a_0, a_1, \ldots, a_{n-1})$ and $(1-a_0, 1-a_1, \ldots, 1-a_{n-1})$ represents the same element $A$. So, the basis $[1, \beta, \beta^2, \ldots, \beta^{n-1}]$ is called redundant basis for any subfield of $K^{(n)}$. Note that the elements in the redundant basis form a cyclic group of order $n$ and

$$\beta \cdot \beta^i = \begin{cases} \beta^{i+1} & i \neq n-1, \\ 1 & i = n-1. \end{cases}$$

We are particularly interested in the following case: Let $K$ be the binary field $\mathbb{F}_2$ and $K^{(n)}$ be a cyclotomic field that $\mathbb{F}_{2^m}$ can be embedded in. The following lemma characterizes the relationship between $m$ and $n$.

**Lemma 1.** [7] Let $n$ be an odd positive integer. Then, $\mathbb{F}_{2^m}$ is contained in $\mathbb{F}_{2^{(n)}}$ if and only if $m$ divides the multiplicative order of $2 \mod n$.

**B. Redundant Basis Multiplication in $\mathbb{F}_{2^m}$**

Consider the redundant basis in $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$:

$$I = [1, \beta, \beta^2, \ldots, \beta^{n-1}].$$

Let field elements $A, B \in \mathbb{F}_{2^m}$ to be represented with respect to $I$ as :

$$A = \sum_{i=0}^{n-1} a_i \beta^i, \ B = \sum_{i=0}^{n-1} b_i \beta^i$$

where $a_i, b_i \in \mathbb{F}_2, i = 0, 1, \ldots, n-1$. Note that $n \geq m + 1$ and $\beta^n = 1$. Then it follows

$$\beta^i \cdot B = b_0 \beta^i + b_1 \beta^{i+1} + \cdots + b_{n-i} \beta^{i-1} + b_{n-1} \beta^{n-1} = \sum_{j=0}^{n-1} b_{(j-i)} \beta^j$$

where $(j-i)$ denotes that $j-i$ is to be reduced modulo $n$. Then the product of field elements $A$ and $B$ can be given by

$$A \cdot B = \sum_{i=0}^{n-1} a_i (\beta^i \cdot B) = \sum_{j=0}^{n-1} \left( \sum_{i=0}^{n-1} a_i b_{(j-i)} \right) \beta^j.$$  

If we define $AB = C \triangleq \sum_{j=0}^{n-1} c_j \beta^j$, then $c_j$ can be given by

$$c_j = \sum_{i=0}^{n-1} a_i b_{(j-i)}, \ j = 0, 1, \ldots, n-1. \quad (2)$$

**Lemma 2.** Assume that $I = [1, \beta, \beta^2, \ldots, \beta^{n-1}]$ is a redundant representation basis for $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$, let $A \in \mathbb{F}_{2^m}$ and $A = (a_0, a_1, \ldots, a_{n-1})$ with respect to $I$, where $n = km + 1$. Assume that $k \geq 2$ and is even. Then,

$$a_k = a_{n-k}, \ k = 1, 2, \ldots, n - 1 \quad (3)$$

Proof: This is a direct result from Lemma 2 in [4] by noting that the redundant basis $I$ and the basis $I_4$ used in [4] satisfy $I = 1 \cup I_4$.

**III. PROPOSED MULTIPLICATION USING REDUNDANT BASIS REPRESENTATION**

When $n = km + 1$ and $k$ is an even number, the complexities of redundant basis multiplier can be reduced further by utilizing the result in Lemma 2.

To facilitate multiplication of elements in the redundant basis representation, a new function $s(i)$ mapping the set of integers to the set $\{0, 1, \ldots, \frac{n-1}{2}\}$ is defined as follows

$$s(i) = \begin{cases} i \ mod \ n & 0 \leq i \leq \frac{n-1}{2} \\ n-i \ mod \ n & \text{Otherwise}. \end{cases}$$

Taking into account (3) and following (2), the product coefficient $c_j, 0 \leq j \leq \frac{n-1}{2}$, can be rewritten as

$$c_j = \sum_{i=0}^{n-1} a_i b_{(j-i)}$$

$$= a_0 b_j + \sum_{i=1}^{n-1} a_s(i) b_{s(j-i)}$$

$$= a_0 b_j + \sum_{i=1}^{n-1} a_s(i) b_{s(j-i)} + \sum_{i=1}^{n-1} a_s(i) b_{s(j+i)}$$

$$= a_0 b_j + \sum_{s(i) = j} a_s(i) [b_{s(j-i)} + b_s(j+i)] \quad (4)$$

From (4) it is easy to prove that

$$c_{n-j} = c_j, \ j = 1, 2, \ldots, \frac{n-1}{2}.$$  

The proof is as follow:

$$c_{n-j} = a_0 b_{n-j} + \sum_{i=1}^{n-1} a_s(i) [b_{s(n-j-i)} + b_{s(n-j+i)}]$$

$$= a_0 b_j + \sum_{i=1}^{n-1} a_s(i) [b_{s(-j-i)} + b_{s(-j+i)}]$$

It is obvious from the definition of the $s(i)$ function that $s(0) = 0$ and $s(i) = s(n-i) = s(-i)$ hence

$$c_{n-j} = a_0 b_j + \sum_{i=1}^{n-1} a_s(i) [b_{s(j+i)} + b_{s(j-i)}]$$

$$= c_j$$
The multiplication complexity from (4), can be reduced even further by using the following lemma.

**Lemma 3.** [4] Assume that \( I = [1, \beta, \beta^2, \ldots, \beta^{n-1}] \) is a redundant representation basis for \( \mathbb{F}_{2^m} \) over \( \mathbb{F}_2 \). Let \( A \in \mathbb{F}_{2^m} \) and \( A = (a_0, a_1, \ldots, a_{n-1}) \) with respect to \( I \), then

\[
A = a_0 + a_1 \beta + \cdots + a_{n-1} \beta^{n-1} = (1 + a_0) + (1 + a_1) \beta + \cdots + (1 + a_{n-1}) \beta^{n-1} \tag{5}
\]

Proof: In redundant basis representation \( \beta^n = 1 \), so \( 1 + \beta + \beta^2 + \cdots + \beta^{n-1} = 0 \).

Taking into account (5), we can assume that \( a_0 = 0 \) (if not use \( A \) complement instead of \( A \)). From (5), the product coefficient \( c_j \) can be given as:

\[
c_j = \sum_{i=1}^{n-1} a_{s(i)} [b_{s(j-i)} + b_{s(j+i)}] \tag{6}
\]

**IV. SERIAL-IN PARALLEL-OUT MULTIPLICATION USING REDUNDANT REPRESENTATION**

**A. Bit-level Serial-In Parallel-Out Architecture**

From (6), a new architecture for bit-level serial-in parallel-out multiplication when \( n = km + 1 \) and \( k \) is even is proposed. The architecture for an \( n \) bit multiplier is shown in Fig 1. It takes \( \frac{n+1}{2} \) clock cycles for the multiplier to finish one operation.

The architecture is made out of one circular shift register which should be initialized with one of the input coefficients. The other input coefficients should enter multiplier in a serial fashion. The top row of XOR gates in the figure are implementing the “+” operation in (6). The bottom row of XOR gates in addition to the output registers create one bit accumulators which implement the summation in (6). The AND gates are to implement one bit multiplication operations between the two input coefficients.

**B. Complexity Comparison**

The gate counts for the proposed architecture can be easily found from fig. 1. The total number of two-input AND gates is equal to \( \frac{n+1}{2} \). The number of two-input XOR gates from the top row is equal to \( \frac{n-1}{2} \) which is the same case for the number of XOR gates in the bottom row. This makes the total number of two-input XOR gates in the architecture equal to \( n-1 \).

The circular shift register is made out of \( n \) flip-flops. Adding the number of output register flip-flops, \( \frac{n+1}{2} \), one can find the total number of flip-flops in the architecture which is \( \frac{3n+1}{2} \).

Complexity comparison between the proposed architecture and the previously proposed serial-in parallel-out architecture [4], are presented in table 1.

**TABLE I**

AREA-TIMING COMPLEXITY COMPARISON BETWEEN BIT-LEVEL PARALLEL-IN SERIAL-OUT MULTIPLIERS

<table>
<thead>
<tr>
<th>Multiplier</th>
<th># AND</th>
<th># XOR</th>
<th># Clock Cycles</th>
<th>Critical Path Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>PISO [4]</td>
<td>( \frac{n+1}{2} )</td>
<td>( \frac{n-1}{2} )</td>
<td>( n )</td>
<td>( T_A + T_X )</td>
</tr>
<tr>
<td>Proposed</td>
<td>( \frac{n+1}{2} )</td>
<td>( n - 1 )</td>
<td>( \frac{n+1}{2} )</td>
<td>( 2T_A + T_X )</td>
</tr>
</tbody>
</table>
### TABLE II
**Hardware Realization Comparison Between Bit-Level Redundant Basis Multipliers**

<table>
<thead>
<tr>
<th>Multipliers</th>
<th>Area</th>
<th># Clock Cycles</th>
<th>Critical Path Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>PISO [4]</td>
<td>109128 $\mu$m$^2$</td>
<td>653</td>
<td>2.57 ns</td>
</tr>
<tr>
<td>Proposed</td>
<td>127318 $\mu$m$^2$</td>
<td>326</td>
<td>3.42 ns</td>
</tr>
</tbody>
</table>

In this table, the delay of a two-input AND gate has been shown by $T_A$ and the delay for a two-input XOR gate has been shown by $T_X$. As can be seen from the architectural analysis, the proposed architecture has a longer critical path delay but since the required number of clock cycles is less than the previous proposal, in total the multiplication delay is smaller than the previous design.

### V. Hardware Implementation

The proposed multiplier and the similar one proposed in [4] have been realized in hardware. For our experience we used the field size of 163 ($m$) which has the smallest cyclotomic ring of 653 ($n$). The field size $m = 233$ was selected since it is the binary field degree recommended by the National Institute of Standards and Technology (NIST) for ECDSA. In our hardware implementation we used the CMOS .18$\mu$m technology from TSMC. Multipliers were first modeled in a hardware description language (HDL) and then the HDL code was synthesized to hardware by Design Compiler software provided by Synopsys.

Hardware simulation results are summarized in table II. It can be seen from the table that the proposed design takes more area than the previous proposal (by 16%), while it is faster (by 33%) than the previous proposal for the field size used in hardware implementation.

### VI. Conclusions

A new Serial-in Parallel-out multiplier using redundant basis for a class of finite fields is proposed. Architectural complexity of the proposed multiplier and previously proposed architectures were compared and it has been shown that the proposed architecture has smaller critical path delay compared to the previous proposal. The proposed architecture and previous similar proposal were realized in hardware and their speed-area comparison has also shown the advantages of the proposed multiplier.

### References