JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Digit-Level Serial-In Parallel-Out Multiplier Using Redundant Representation for a Class of Finite Fields

MEERZA NOUSIN[1],Dr.B.NAGESHWARRAO[2] ,Dr.T.VAMSHI[3]

[1]M.Tech Student, Talla Padmavathi College of Engineering, Somidi, Kazipet, Telangana, 506003
[2]Assoc Professor, Talla Padmavathi College of Engineering Student, ,Somidi, Kazipet, Telangana, 506003
[3]Assoc Professor, Talla Padmavathi College of Engineering,Somidi, Kazipet, Telangana, 506003

[1]md.mehareen1722@gmail.com,[2]nagesh.south@gmail.com, [3]vamshi22g@gmail.com

Abstract

We offer finite field multipliers with two digits of digits at redundant representations. Performing field multiplication with redundant representation would necessitate extra hardware resources because of the redundancy introduced by embedding in cyclotomic fields. Two new multiplying algorithms and their corresponding designs are suggested to address the problem of repetitive representations in a category of finite fields. In terms of the area-delay product, both of the suggested architectures significantly exceed existing digit-level multipliers based on the same methodology. In addition, it is shown that the suggested multipliers outperform other newly proposed optimum normal basis multipliers in regards of area-delay complexity for a subset of fields. Additional information is provided on the key aspects of the proposed multipliers' postplace&route application-specific integrated circuit implementations for three realistic digit size options

Key words: Digit-level architecture, finite field arithmetic, multiplication algorithm, redundant representation

1. Introduction

Recently, finite field computations have been receiving increasing interest because of its variety of applications in encoding, error control codes, and cryptography. Two of the three most well-known methods of encryption, ElGamal and ECC, use finite field arithmetic. Using the underlying finite field, it is possible to execute finite field computations. With field multiplication, it is possible to do more difficult operations like field exponentiation & field inverting without having to resort to the simpler field operations.

In finite field arithmetic, the theory of representing bases is utilised to represent field constituents in a manner similar to that found in linear algebra. Because of the hardware and cryptosystem requirements, a computer's performance can be greatly impacted by the quality of representation system. Polynomial basis (PB), normal basis (NB), redundant basis (RB), and dual basis (DB) are some of the representation techniques for extended binary fields that have been presented in literature. The squaring operation can be accomplished in both the NB and redundant representations by performing a simple permutation on the coordinates. This provides better performance for implementations of encryption algorithm that frequently use squaring or algebraic, such as ECC point addition/doubling. Furthermore, the unique feature of redundant representation in supporting ring-type operations makes it of particular relevance. In addition to being practically free, this method eliminates the requirement for modularity reduction in multiplication, making it even more efficient. For NB multiplication, Gao et al. proposed the idea of integrating a field inside a larger ring.

This representation method was then used by Wu and colleagues to develop finite field multiplication, which is known as RB. Several architectures, such as comb-style construction and LFSR-based architectures, have since been presented in an attempt to speed up multiplication or simplify the circuitry. Decomposition of serial/parallel structures in terms of their area, time, and power complexity has been proposed in the last few years by the team of Xie and colleagues. It is not a one-to-one mapping procedure when embedding an m-by-n-by-n-by-n cyclotomic field, which is the fundamental downside of redundant representation in spite of the architecture used. Thus, it takes more ascii characters a field element, and the number of bits depends on how large that cyclotomic field where the field is embedded is. The focus of this paper is on RB multiplier digit-level architectures. We show that a special property of redundant information can be leveraged to greatly reduce the takes a lot of effort of RB amplifiers to correct for the inherent redundancies in just this representation system. In this paper, two different multiplication algorithms and architectures are discussed. For hardware implementation, it's been shown that the proposed designs have extremely regular structures. Both the suggested architecture outperformed other RB architectures when measured by area-delay product in comparison to existing digit-level RB architectures. Additionally, the proposed multipliers' results are contrasted with those of many other, more optimal NB (ONB) multipliers. Finally, the hardware implementations of the recommended multipliers for three feasible digit sizes are provided.

2. Literature survey

“Finite Field Multiplier Using Redundant Representation”

Configurations for finite field multiplication employing redundant representation are presented in this paper. Cyclotomic rings have an elegant multiplicative structure, therefore the main idea is to embed an infinite field within one of these rings. For example, we can construct the multiplication in a hybrid/partial-parallel manner thanks to the area-time tradeoffs provided by our architectures. The VLSI implementation of this hybrid architecture in very large fields is significant. It is a simple permutation of the coordinates to perform the square function using the redundant representation. The suggested bit-serial and hybrids multiplier designs have low space complexity when an optimum normal foundation is in place. As an alternative to employing redundant representations, constant multiplication is suggested.

“A New Finite-Field Multiplier Using Redundant Representation”

The use of redundancy representation in a new serial-in concurrent finite field multiplier is proposed. The suggested design is shown to have either a lower complexity and a comparable critical path delay, or a lower critical path delay and a comparable complexity, when compared to other architectures utilising the same representation. If a type I optimum normal basis exists, the proposed multiplier performs better than regular basis multipliers. This document also includes a digit-level version of the new multiplier.

“Efficient Implementation of Finite Field Multipliers over Binary Extension Fields”

Symmetric-key and public-key cryptography are two fundamentally different kinds of cryptography (also known as asymmetric-key). In contrast to symmetric-key cryptography, which relies on a single secret that is known only by the transmitter and recipient, public-key cryptography uses two different but mathematically linked keys to ensure the integrity of the security mechanism. Finite field arithmetic-based public-key cryptosystems, such as EC and ElGamal, are two instances of this type. For public-key cryptography, the Elliptic Curve (EC) technique is the most efficient. With EC cryptosystems, you may achieve the same level of security with a smaller key size, making them ideal for many applications. In contrast to non-public-key cryptosystems, it is compute intensive and consumes a lot of electricity. Operation hierarchies in cryptosystems are generally characterised in terms of finite field math operations as the bottom layer of the hierarchical structure. All cryptosystems relying on finite field arithmetic use finite field multiplication because it is both computationally difficult and one of the most commonly used finite operations. It is from a hardware integration perspective that this dissertation primarily focuses on the efficient computing and development of finite field multiplication

3. Related work

3.1 Redundant Representation for F2m

Let us designate F2 as a field with the characteristic value of 2, and xn 1 as a polynomial with degree equal to or greater than n over F2. Then, the field that splits xn into two halves, which is denoted by, is referred to as the n - th cyclotomic fields over F2. Let serve as an example of a primitive n - th component of unity that exists in an extensions field of F2. After that, F(n)2 is produced when is applied to F2, and the components of can be recast in the form of.

$$A = a_0 + a_1\beta + a_2\beta^2 + \cdots + a_{n-1}\beta^{n-1}, a_i \in \mathbb{F}_2. \qquad (1)$$

This particular representation of A is not the only one possible; more specifically, for any element of A that is represented by an n-tuple with the form (a0, a1, • • •, an1), where ai is a positive integer between 0 and F2, there are multiple tuples that can also be used to represent the same element. As an illustration, each element in and the ones' complement of that element both indicate the same field element, as described in Lemma 1.

Lemma 1: Assume that the field element E is denoted by (e0, e1,..., en1), and that ei F2 is defined with regard to I = 1,,.., n1. Then

$$\begin{aligned} E &= e_0 + e_1\beta + \cdots + e_{n-1}\beta^{n-1} \\ &= (1+e_0) + (1+e_1)\beta + \cdots + (1+e_{n-1})\beta^{n-1}. \qquad (2) \end{aligned}$$

3.2 Multiplication Using Redundant Representation in $\mathbb{F}_{2^m}$

In finite field arithmetic, utilising RB offers a number of distinct benefits, one of which is the elimination of the requirement for modular reduction in the operation of multiplication. This important property derived from the fact that now the basis elements 1,, 2,..., n1 form a cyclical group of order n. n denotes the number of cycles in the group. As a direct consequence

$$\beta \cdot \beta^i = \begin{cases} \beta^{i+1} & i \neq n-1 \\ 1 & i = n-1. \end{cases} \qquad (3)$$

Let the field components A and B F2m be represented with reference towards the RB I = -1, -1, -2,..., n-1 as:

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

correspondingly, with the exception that Take note that n is greater than m + 1 and that n = 1. Then the answer to how to get C, the combination a And B, is:

$$\begin{aligned} C = A \cdot B &= \sum_{i=0}^{n-1} (a_i \beta^i) \cdot B \\ &= \sum_{i=0}^{n-1} a_i \left(\sum_{j=0}^{n-1} b_j \beta^{i+j} \right) \\ &= \sum_{i=0}^{n-1} a_i \left(\sum_{j=0}^{n-1} b_{(j-i)_n} \beta^j \right) \\ &= \sum_{j=0}^{n-1} \left(\sum_{i=0}^{n-1} a_i b_{(j-i)_n} \right) \beta^j \qquad (4) \end{aligned}$$

$$c_j = \sum_{i=0}^{n-1} a_i b_{(j-i)_n}, \quad j = 0, 1, \ldots, n-1. \qquad (5)$$

4. Proposed system

In this portion of the study, a novel RB multiplication approach is introduced. Due to the success of this technology, two new digit level SIPO architectures are presented for serial-in parallel-out (SIPO).

These architectures are employed for a category of finite fields in which n can be expressed with n = Tm + 1, where T 2 & is an even number. In the next part, we'll see how this limitation aids in the development of an architecture that reduces the multiplier's complexity. Because of this, Corollary 1 describes a specific property. Consider that redundant representations is given by (a0,a1,..,an1) with regard to Rb I over F2m as the first corollary. Assuming that T 2 and n is even, it can be written as Tm + 1.

$$a_k = a_{n-k}, \quad k = 1, 2, \ldots, n-1. \qquad (6)$$

It is possible to calculate the degree of the lowest cyclotomic field included.

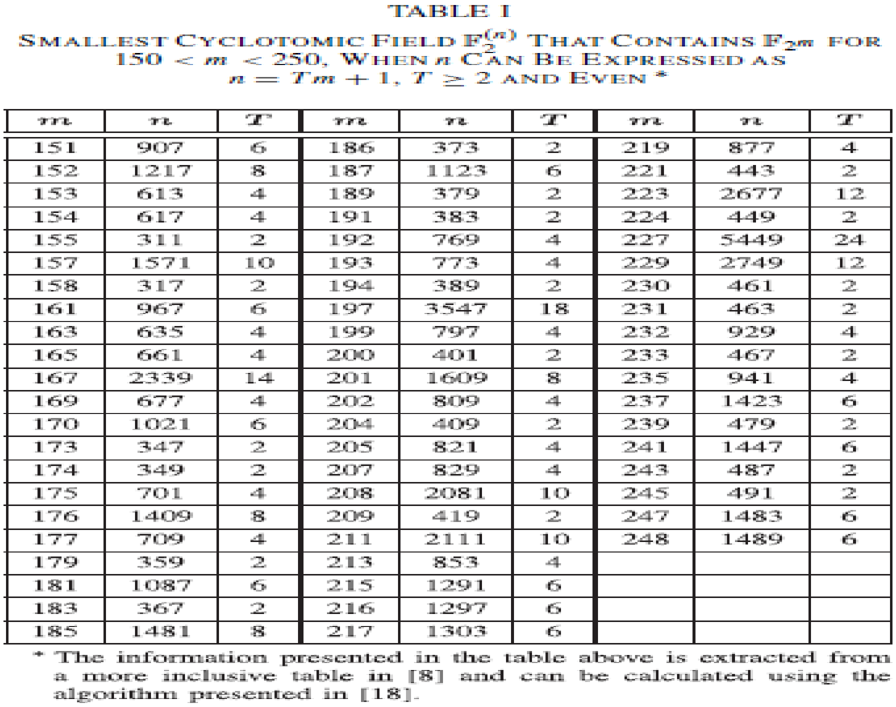


TABLE I

SMALLEST CYCLOTOMIC FIELD $F_2^{(n)}$ THAT CONTAINS F_{2^m} FOR $150 < m < 250$, WHEN n CAN BE EXPRESSED AS $n = Tm + 1$, $T \geq 2$ AND EVEN *

m	n	T	m	n	T	m	n	T
151	907	6	186	373	2	219	877	4
152	1217	8	187	1123	6	221	443	2
153	613	4	189	379	2	223	2677	12
154	617	4	191	383	2	224	449	2
155	311	2	192	769	4	227	5449	24
157	1571	10	193	773	4	229	2749	12
158	317	2	194	389	2	230	461	2
161	967	6	197	3547	18	231	463	2
163	635	4	199	797	4	232	929	4
165	661	4	200	401	2	233	467	2
167	2339	14	201	1609	8	235	941	4
169	677	4	202	809	4	237	1423	6
170	1021	6	204	409	2	239	479	2
173	347	2	205	821	4	241	1447	6
174	349	2	207	829	4	243	487	2
175	701	4	208	2081	10	245	491	2
176	1409	8	209	419	2	247	1483	6
177	709	4	211	2111	10	248	1489	6
179	359	2	213	853	4			
181	1087	6	215	1291	6			
183	367	2	216	1297	6			
185	1481	8	217	1303	6			

* The information presented in the table above is extracted from a more inclusive table in [8] and can be calculated using the algorithm presented in [18].

Field sizes of the following are recommended for use in ECC applications:picked between 150 and 600 based on security standards [19]. Over 60% of any and all finite fields in the practical domain are covered by Corollary 1 thru [18]. Table I lists the orders of the shortest cyclotomic fields that satisfy the criterion of Corollary 1 for the first 100 field in the given range.

which can be employed.

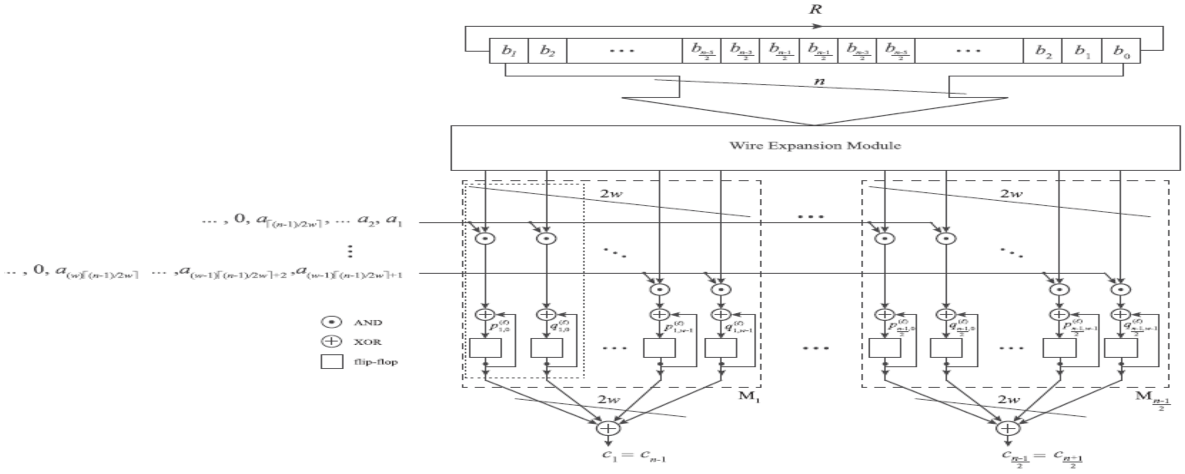

R
b_l
b_2
...
$b_{\frac{n-5}{2}}$
$b_{\frac{n-3}{2}}$
$b_{\frac{n-1}{2}}$
$b_{\frac{n-1}{2}}$
$b_{\frac{n-3}{2}}$
$b_{\frac{n-5}{2}}$
...
b_2
b_1
b_0
n
Wire Expansion Module
2w
2w
$\ldots, 0, a_{\lceil (n-1)/2w \rceil}, \ldots a_2, a_1$
$\ldots, 0, a_{(w)\lceil (n-1)/2w \rceil} \ldots, a_{(w-1)\lceil (n-1)/2w \rceil+2}, a_{(w-1)\lceil (n-1)/2w \rceil+1}$
AND
XOR
flip-flop
$p^{(i)}_{1,0}$
$q^{(i)}_{1,0}$
$p^{(i)}_{1,w-1}$
$q^{(i)}_{1,w-1}$
$p^{(i)}_{\frac{n-1}{2},0}$
$q^{(i)}_{\frac{n-1}{2},0}$
$p^{(i)}_{\frac{n-1}{2},w-1}$
$q^{(i)}_{\frac{n-1}{2},w-1}$
2w
M_1
2w
$M_{\frac{n-1}{2}}$
$c_1 = c_{n-1}$
$c_{\frac{n-1}{2}} = c_{\frac{n+1}{2}}$

Fig. 1. Proposed architecture for digit-level SIPO RB multiplier, DL-SRB-*a*.

Step 5 requires the coefficients generated in Step 2 to be exact. From left to right, bn1, bn2,..., b0, should be loaded into this circular shift register. There's no such thing as a "increasing" variable in the function b(j+kd+_) in Step 6. It is necessary to use a comparable cyclical shift register, namely R2, with the identical initial values but also with the opposing shift direction to create the appropriate coefficients.

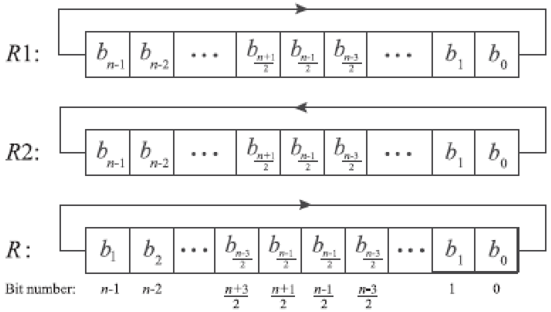

R1:
b_{n-1} b_{n-2} ... $b_{\frac{n+1}{2}}$ $b_{\frac{n-1}{2}}$ $b_{\frac{n-3}{2}}$ ... b_1 b_0
R2:
b_{n-1} b_{n-2} ... $b_{\frac{n+1}{2}}$ $b_{\frac{n-1}{2}}$ $b_{\frac{n-3}{2}}$ ... b_1 b_0
R:
b_1 b_2 ... $b_{\frac{n-3}{2}}$ $b_{\frac{n-1}{2}}$ $b_{\frac{n-1}{2}}$ $b_{\frac{n-3}{2}}$ ... b_1 b_0
Bit number: n-1 n-2 $\frac{n+3}{2}$ $\frac{n+1}{2}$ $\frac{n-1}{2}$ $\frac{n-3}{2}$ 1 0

Fig. 2. Circular *n*-bit shift register to store coordinates of operand *B*.

Operand B's position in the shift register is represented by the n-bit circular shift register shown in this figure.To put it another way, the proposed design is a sequential circuit accompanied by such a combinational circuit. Way of summarizing p(l) j,k & q(l) j,k are iteratively computed in the sequential component of the circuit, which includes the XOR tree.

At the end of each clock cycle, the flip-flops are filled with data. Because the XOR trees' output has no bearing on the sequential circuit's computations throughout the first d clock pulse, they do not need to be saved. However, after d clock cycles, the resultant dimensions will not be available. The binary tree of (2w 1) two-input Two input (combinational circuit) has an additional delay of [log2 2w]TX before the product dimensions can be read from either the output end. It is recommended that this step be completed in multicycles to prevent the series circuit from becoming critical path Using intermediate flip-flops to split a long trip into smaller chunks is a frequent method. Flip-flops can be avoided if the combinational circuit's inputs remain constant such that the circuitry has sufficient time to generate legitimate outputs. Specifically, this is accomplished by adding dex zeroes to the beginning of each input sequence starting with A0, going through A1, and ending with Aw1, in the proposed architecture. dex is the number of additional clock cycles required once Steps 5 and 6 are complete. Calculating dex can be done using

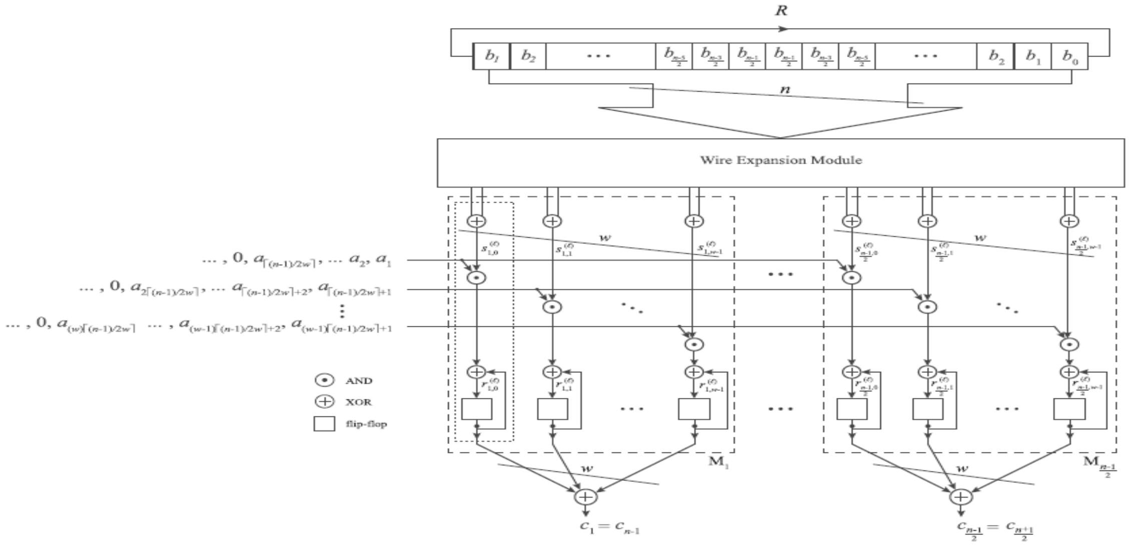
$$d_{\mathrm{ex}} = \left\lceil \frac{\lceil \log_2 2w \rceil T_X}{T_{\mathrm{clock}}} \right\rceil \qquad (19)$$

There is a period of time known as Tclock. if the critical path latency is equal to the clock period, Tclock should indeed be exchanged with Tcp in the design (19). Finally, d + dex is the sum of all the clock cycles required to perform a single multiplication operation.

4.1 New Multiplier Architecture, DL-SRB-b

The volume of logic circuits and flip-flops utilised in the design depicted in Fig. 1 can be greatly decreased at the cost of a minor increase in critical path time. Define two intermediary outputs s(l) j,k & r (l) j,k and k = 0 with 1 for w = 1, 2 and so on as indicated in (14), rather then the decomposition stated in (14). This results in the closed formula (13).

$$\begin{cases} s_{j,k}^{(\ell)} = [b_{\varphi(j-kd-\ell)} + b_{\varphi(j+kd+\ell)}] \\ r_{j,k}^{(0)} = 0 \text{ and } r_{j,k}^{(\ell)} = r_{j,k}^{(\ell-1)} + \hat{a}_{(kd+\ell)} s_{j,k}^{(\ell)}. \end{cases} \qquad (20)$$



\

Fig. 3. Proposed architecture for digit-Level SIPO RB multiplier, DL-SRB-*b*.

It holds the values of signal r (d)j,k after the d clock equivalent to

$$r_{j,k}^{(d)} = \sum_{\ell=1}^{d} \hat{a}_{(kd+\ell)}[b_{\varphi(j-kd-\ell)} + b_{\varphi(j+kd+\ell)}]. \qquad (21)$$

Product cartesian coordinate ci can be represented in terms of (20) and (13).

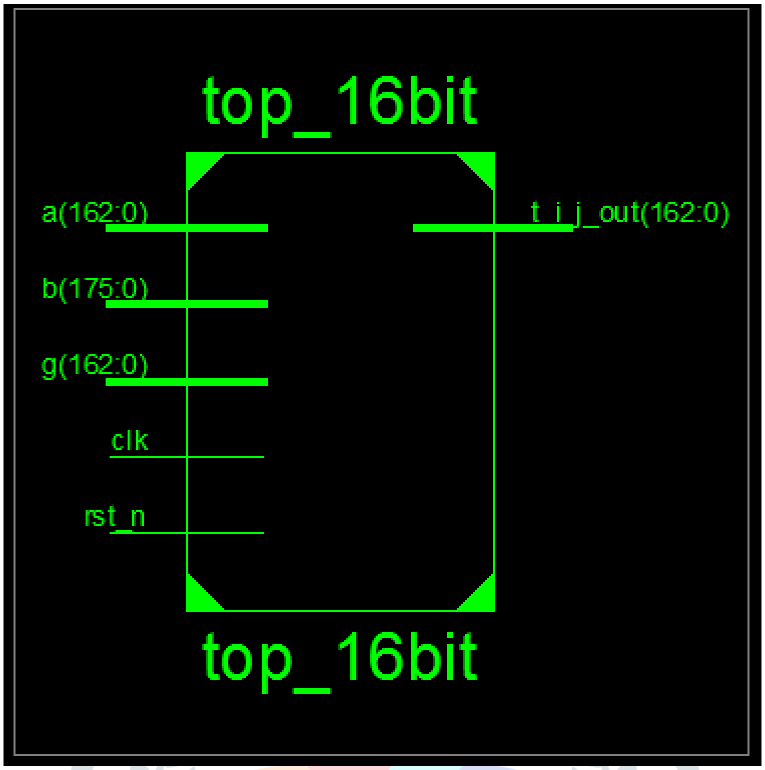
$$c_j = \sum_{k=0}^{w-1} r_{j,k}^{(d)}. \qquad (22)$$

D and dex are two elements of the DL-SRB-b multiplication delay, same like in DL-SRB-a. In the first half, we look at the effects of modules Mj throughout d clock cycles on Stages 5 and 6 of both the method. Part two is the lag time of an input XOR gate or the binary tree of two-input XOR gates that has (w 1) inputs. Figure out the amount of clock cycles it takes to accomplish one multiplication by assuming two-input XOR gates are employed.

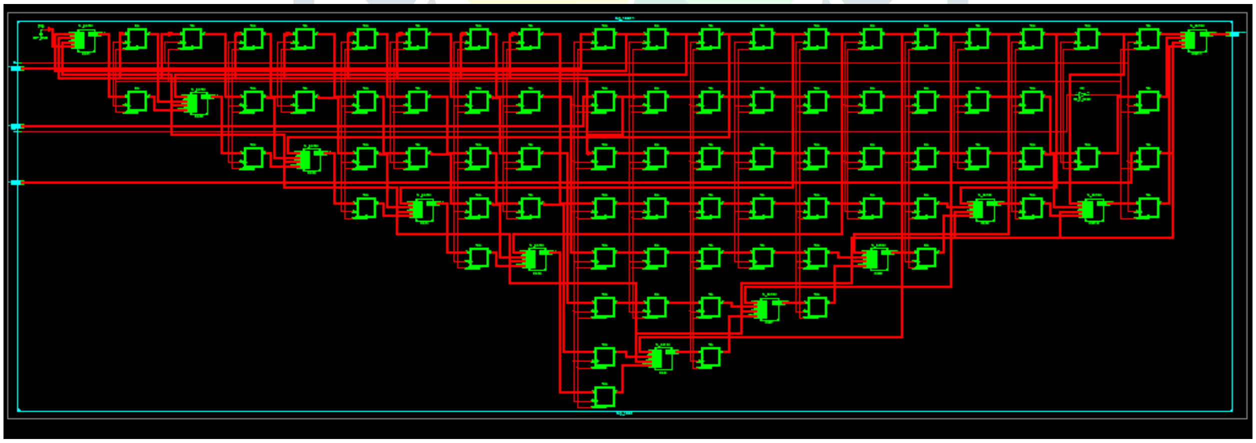
$$d + \left\lceil \frac{\lceil \log_2 w \rceil T_X}{T_{\text{clock}}} \right\rceil. \qquad (23)$$

5. Results

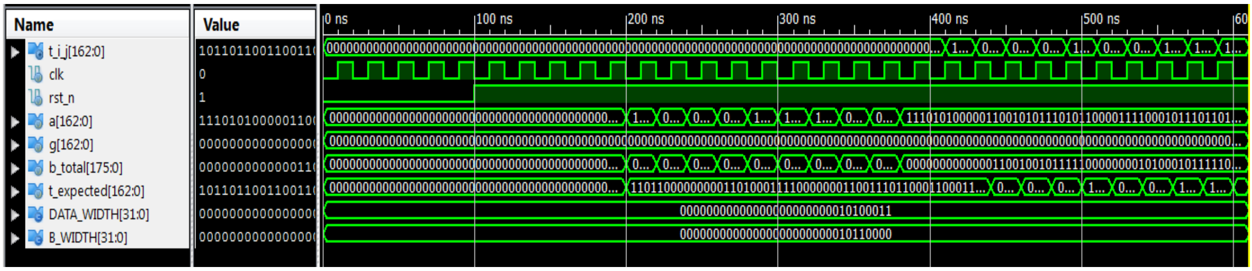
Entity diagram:



RTL schematic:



Simulation results:



```
Name                 Value
t_i_j[162:0]         101101100110011(
clk                  0
rst_n                1
a[162:0]             111010100000110(
g[162:0]             000000000000000(
b_total[175:0]       000000000000011(
t_expected[162:0]    101101100110011(
DATA_WIDTH[31:0]     000000000000000(
B_WIDTH[31:0]        000000000000000(
```

6. Conclusion

Using redundancy representation, two novel digit-level SIPO finite-field multipliers have been developed. According to this formula, for around 60 percent of the measured values within the reasonable number for ECC purposes, the least cyclotomic field size that may be embedded (n) is n = Tm + 1 when the extension degrees is even and larger or equal to 2. The repetition problem in this representations was alleviated by utilising a unique aspect of redundant representation. Analyzing numerical complexity, it was shown that both novel architectures had significantly lower delay costs when compared to the current RB designs. When area-delay complexity was used as a performance measure, one of the recommended architectures outperformed the most relevant RB architecture by at least 2.12 times (for various digit sizes over F2233). When T = 2, the proposal can outperform ONB multipliers in around 20% of cases and outperform NB multipliers in cases when there is no ONB but T = 2. (e.g., field sizes 200, 204, and 224). Furthermore, a 65-nm CMOS VLSI execution of the proposed structures with binary extension field 233 and 3 different digit sizes was provided.

7. References

[1] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Sep. 2006.

[2] I. F.Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography* (London Mathematical Society Lecture Note Series). Cambridge, U.K.: Cambridge Univ. Press, 1999.

[3] A. J. Memezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography* (Discrete Mathematics and Its Applications). Boca Raton, FL, USA: CRC Press, 1996.

[4] T. Itoh and S. Tsujii, “A fast algorithm for computing multiplicative inverses in *GF(2m)* using normal basis,” *Inf. Comput.*, vol. 78, no. 3, pp. 171–177, 1988.

[5] C. Rebeiro, S. Roy, D. Reddy, and D. Mukhopadhyay, “Revisiting the Itoh–Tsujii inversion algorithm for FPGA platforms,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1508–1512, Aug. 2011.

[6] E. D. Mastrovito, “VLSI architectures for computations in Galois fields,” Ph.D. dissertation, Dept. Electr. Eng., Linköping Univ., Linköping, Sweden, 1991.

[7] J. Omura and J. Massey, “Computational method and apparatus for finite field arithmetic,” U.S. Patent 4 587 627, May 6, 1986.

[8] H. Wu, M. A. Hasan, I. F. Blake, and S. Gao, “Finite field multiplier using redundant representation,” *IEEE Trans. Comput.*, vol. 51, no. 11, pp. 1306–1316, Nov. 2002.

[9] D. Jungnickel, A. J. Menezes, and S. A. Vanstone, “On the number of self-dual bases of *GF(qm)* over *GF(q)*,” *Proc. Amer. Math. Soc.*, vol. 109, no. 1, pp. 23–29, 1990.

[10] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup, “Algorithms for exponentiation in finite fields,” *J. Symbolic Comput.*, vol. 29, no. 6, pp. 879–889, 2000.

[11] S. Gao, J. von zur Gathen, and D. Panario, "Gauss periods and fast exponentiation in finite fields," in *LATIN Theoretical Informatics* (Lecture Notes in Computer Science), vol. 911. Berlin, Germany: Springer, 1995, pp. 311–322.

[12] A. H. Namin, H. Wu, and M. Ahmadi, "Comb architectures for finite field multiplication in *(Fm*2 *)*," *IEEE Trans. Comput.*, vol. 56, no. 7, pp. 909–916, Jul. 2007.

[13] A. H. Namin, H. Wu, and M. Ahmadi, "A new finite-field multiplier using redundant representation," *IEEE Trans. Comput.*, vol. 57, no. 5, pp. 716–720, May 2008.

[14] A. H. Namin, H. Wu, and M. Ahmadi, "An efficient finite field multiplier using redundant representation," *ACM Trans. Embedded Comput. Syst.*, vol. 11, no. 2, Jul. 2012, Art. no. 31.

[15] J. Xie, P. Meher, and Z.-H. Mao, "High-throughput finite field multipliers using redundant basis for FPGA and ASIC implementations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 1, pp. 110–119,Jan. 2015.

[16] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 1997.

[17] D. W. Ash, I. F. Blake, and S. A. Vanstone, "Low complexity normal bases," *Discrete Appl. Math.*, vol. 25, no. 3, pp. 191–210, 1989.

[18] H. Wu, M. Hasan, and I. Blake, "Highly regular architectures for finite field computation using redundant basis," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 1717, C. K. Koç and C. Paar, Eds. Berlin, Germany: Springer, 1999,pp. 269–279.

[19] C. F. Kerry and P. D. Gallagher, "Digital signature standard DSS," U.S. Dept. Commerce,
Nat. Inst. Standards Technol. Tech. Rep. FIPS 186-4, Jul. 2013. [Online]. Available: http://csrc.nist.gov/publications/PubsFIPSArch.html

[20] R. Azarderakhsh and A. Reyhani-Masoleh, "Low-complexity multiplier architectures for single and hybrid-double multiplications in Gaussian normal bases," *IEEE Trans. Comput.*, vol. 62, no. 4, pp. 744–757, Apr. 2013.