IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 4, Ver. I (Jul. - Aug. 2016), PP 35-45
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Implementation of High-Throughput Digit-Serial Redundant Basis Multipliers over Finite Field

Jyothi Leonore Dake[1], Sudheer Kumar Terlapu[2]
[1](M.Tech-VLSID, Department of Electronics and Communication Engineering,Shri Vishnu Engineering College for Women (Autonomous), India)
[2](Associate Professor,Department of Electronics and Communication Engineering, Shri Vishnu Engineering College for Women (Autonomous), India)

Abstract*:In elliptical curve cryptography the redundant basis (RB) multipliers for finite field have achieved immense popularity. The high-throughput multipliers are presented and they utilize redundant representation. In this paper, a novel recursive decomposition algorithm is presented for digit-level RB multiplication to obtain digit-serial implementation. The signal-flow graph (SFG) is extended to obtain the processor- space flow graph (PSFG) and also to acquire the three novel multipliers. Implementation of 10 bit digit-serial RB multipliers is presented in this work. The proposed structures are simulated and synthesized in Xilinx 12.2 using Verilog HDL.*

Keywords: *Cryptography, Digit-serial multiplication, finite field, redundant representation.*

I. Introduction

Finite field GF (2^m) is a field that contains finitely many fields. It is especially useful in translate computer data, which present in the binary form. Finite Field has wide applications in cryptography and error control coding [1], [2]. The key arithmetic unit for multiple systems based on computations of finite field is finite field multiplier because the complex operations like division and inversioncan be broken down into successive multiplication operation. The most common arithmetic is multiplication which is useful to obtain efficient multipliers [3].

Both the hardware and software architectures are studied for computing multiplications over finite field [4]. The mostly used bases for finite fields are polynomial (PB), normal (NB), triangular (TB), and redundant (RB) [5]. Basis is a set of vectors that, in a linear combination, can represent every vector in given a vector space. Redundant basis is attractive due to its free squaring and modular reduction for multiplication [7]. A redundant representation is extracted from minimalcyclotomic ring and the arithmetic operation can be performed in the ring by embed the present field [9].

A number of structures have been designed for efficient finite field multiplication over finite field based on RB. Semi-systolic Montgomery multiplier ispresented in [4]. Super-systolic multiplier has been reported by Pramod Kumar Mehar. Bit-Serial/Parallel multipliers [8], Comb style architectures are presented formerly and also several other RB multipliers are designed for hardware efficiency and throughput [6].

In this contribute, an efficient high-throughput digit-serial/parallel multiplier designs over finite field based on RB is presented. A novel recursive decomposition scheme is presented, based on that parallel algorithms are obtained for high-throughput digit-serial multiplication. By depicting the parallel algorithm to a regular two dimensional signal-flow-graph (SFG) array go after by projection of SFG to onedimensional processor-space flow graph (PSFG), the algorithm is mapped to three multiplier architectures. In this work, the implementation of 10-bit digit-serial RB multipliers is presented to obtain high-throughput.

The organization of this paper is as follows: Mathematical representation is presented in section II. High-throughput structures for digit-serial RB multipliers are derived from the proposedalgorithm mentioned in section III. Implementation and Simulation results are presented in section IV. Conclusions are presented in section V.

II. Mathematical Representation

Assume x to be a primitive nth root of unity, components in finite field GF (2^m) are often described within the form:

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

Where $a_i$belongs to GF (2), for $0 \leq i \leq n-1$, alike the set $\{1, x, x^2, \cdots, x^{n-1}\}$ is defined 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 2 is primitive root modulo $(m + 1)$ for a finite field, there beinga type I optimal normal basis (ONB).

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

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

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

Thus a_i,$b_i$belongs to GF (2). Let Aand B are inputs and product is C, is demonstrated as follows:

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

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

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

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

$c_i = \sum_{i=0}^{n-1} b_{(i-j)_n} a_j$(8)

Alternately, we can write (8) in a bit-level matrix-vector form as:

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

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

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

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

……

$B^{n-1} = \sum_{i=0}^{n-1} b_i^{n-1} x^i = b_1 + b_2 x + \cdots + b_0 x^{n-1}$(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 $1 \le s \le n-1$, Let Q and P are two integers alike n = QP + r, where $0 \le r \le P$.For ease, assumer = *0,* and decompose the input operand A into Q number of bit-vectors $A_u$for u= 0, 1,⋯Q-1,as follows:

$A_0 = [a_0 a_Q \cdots a_{n-Q}]$ (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 can produce Q units of shifted vector operands B_u for u= 0, 1, ⋯ ,Q -1, as 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)

… … …

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

The product $C = AB$ which is obtained from (6) are broken down into productsQ of vectors A_u and B_u for u = 0, 1, ⋯ , Q-1 as:

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

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

Where $\overline{C_u}$ denotes:

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

Note that $A_u$for u =0, 1, ⋯ , Q -1 is a *P* point bit – vector. $B_u$for u = 0, 1, ⋯ , Q – 1 is a *P* bit-shifted forms of operand *B.* Based on (21) and (22) proposed digit-serial algorithm is described.

Algorithm for proposed digit-serial RB multiplication

Inputs: A and B are pair of elements in Finite Field GF (2^m) to be multiplied.
Output: C = A .B
Initialization: D = 0
    For u = 0 to Q - 1
    For u = 0 to *P* – 1
D = D +$B_u A_u^T$
    End For
    End For
C = D

III. High-Throughput Structures For Digit-Serial RB Multipliers

3.1Structure – I for Digit-serial RB multiplier

The proposed digit-serial RB multiplier is derived from the SFG of the proposed algorithm. From (21) and (22), the representation of RB multiplication is by two dimensional SFG in Fig.1. The SFG consists of Q number of arrays which are in parallel; each array is with (P-1) bit-shifting nodes which is S node. The S nodes are two types they are S-I and S-II. The one position circular bit-shifting is carryout by S-I and Q positions circular bit-shifting is carryout by S-II. And it also consists of P multiplication nodes and addition nodes, where M nodes and A nodes.

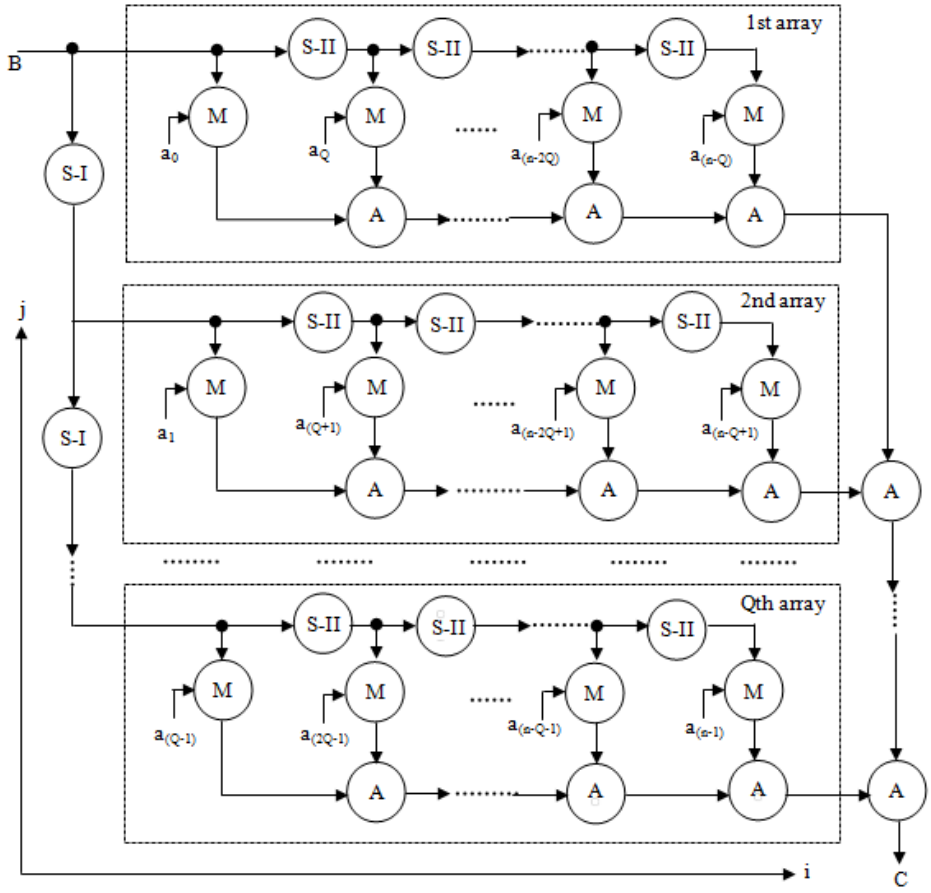
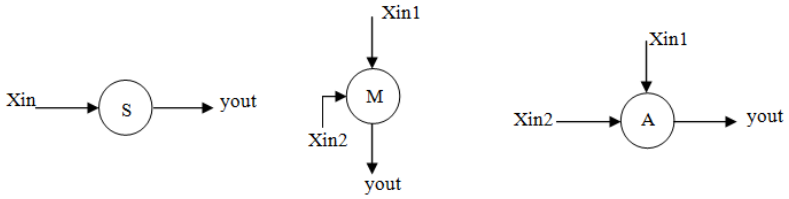


Fig.1. The proposed Signal-flow graph (SFG) for parallel realization of RB multiplication

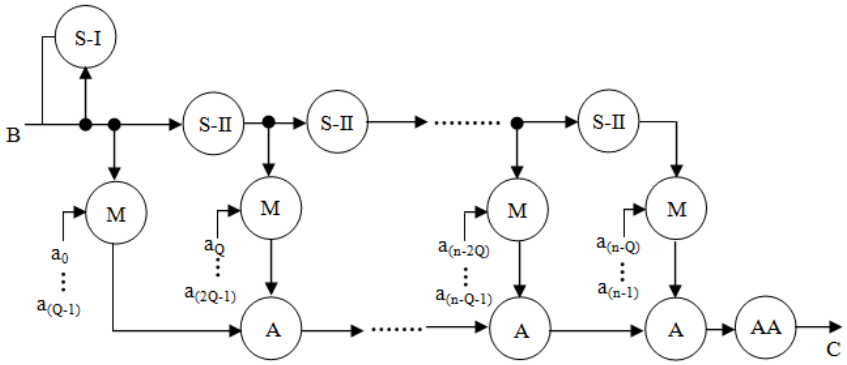
The role of M nodes and A nodes are described in Fig.2 (b) and 2(c). M node carryout AND operation of each serial-input bits of Awith the B input bits by bit-shifting form, and XOR operation is carryout by the each Anode. The final output is obtained by performing the bit by bit XOR operation of the operands Fig.1.By addition of the Q parallel arrays output the required product word is obtained. To obtain the PSFG (Fig.3), the SFG is projected along the jth direction for digit-serial multiplication.In PSFG during each clock cycle the p number of input bits carried in parallel to multiplication node. The PSFG functionality is as same as the SFG inFig.1 It consists of an extra node which is add-accumulation node (AA) and the role of the add-accumulation is to carryout accumulation operation to produce necessary result.



Bit-shifting (Xin) →youtXin1. Xin2→yout Xin1+Xin2→yout

(a) (b) (c)

Fig. 2. (a) Functional representation of S node. (b) Functional representation of M node. (c) Functional Representation of A node



(a)

(b)

For 1≤T ≤ Q;
R← R+ xin; T← T+1
If T=Q then yout← R;
T←0; R←; R← 0; Endif

Fig.3. Processor- space flow graph (PSFG) of digit-serial realization of finite field RB multiplication over GF (2^m). (a) The PSFG. (b) Functional representation of add-accumulation (AA) node.

The digit-serial RB multiplier shown in Fig.4, mentioned as structure-I. Structure-I consists of three blocks, which are bit-permutation module (BPM), partial product generation module (PPGM) and finite fieldaccumulator module. The BPM carries out rewiring of inputsB and the output is fed to the partial product generation unit. The PPGM is with the AND, XOR and register cells which carry out the function of M node. And the finite field accumulator block consistent with n-bit parallel accumulation units. The recent input which is received is added with the past accumulated result, and the sum is retain in the register cell and used in the next cycle. And successive output is obtained. Fig.7 shows the structure of partial product generation module which consists of XOR cell, AND cell and register cells with n parallel input bits and n parallel output bits.

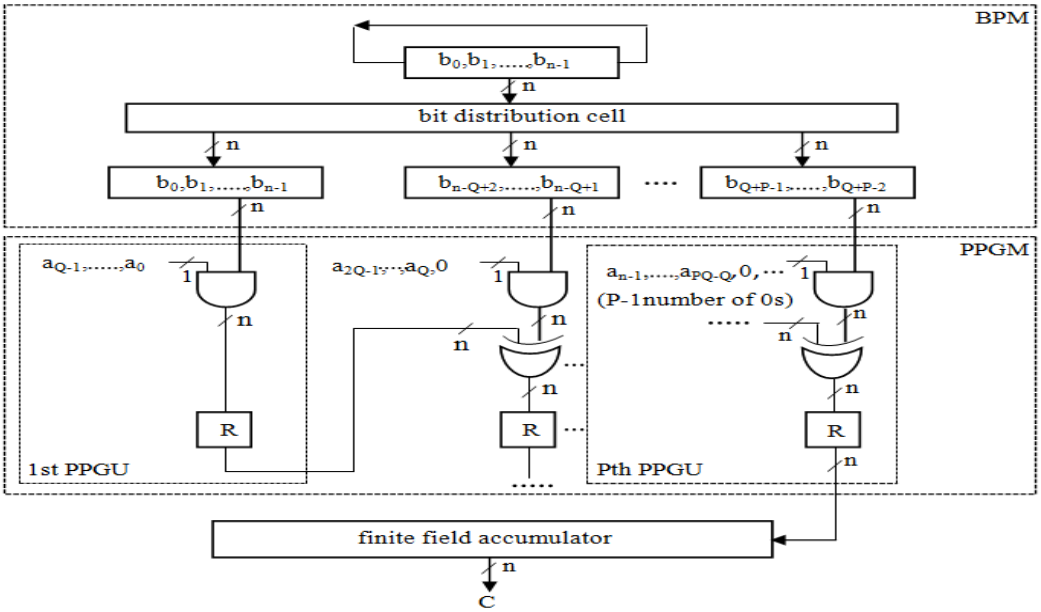


Fig.4. structure-I for digit-serial RB multiplier

3.2 Modification of Structure –I for Digit-Serial RB Multiplier

We can have (*P=kd+l),* for any p integer value, where 0≤ *l* <*d* and *d*<*P*. For simpleness, we assume l=0, however can easily extended to the cases where $l \neq 0$. Define 0≤ $h \leq k-1$, and 0≤ $f \leq d-1$, such that (22) can be as:

$$\overline{C_u}=\sum_{h=0}^{k-1}\sum_{f=0}^{d-1} B^{u+fhQ}\ a_{u+fhQ} \qquad (23)$$

By depending on the (23), the PSFG is modified to obtain appropriate digit-serial multiplier structure Fig.6, a set of shifting nodes, a set of multiplication nodes and a set of addition nodes of PSFG are combined to form overall node. And these nodes are executed by new PPGU to obtain PPGM of P/2 PPGUs. Suitably, in the structure of Fig.4 the two PPGU are appeared into a new PPGU, and it consists of two AND cells, two XOR cells and it needs only one XOR cell at the first PPGU of the structure-I when d=2. The functionality of the AND, XOR and register cells are same as the structure-I in Fig.4.

3.3Structure- II for digit-serial RB multiplier

The Structure-II for digit-serial RB multiplier is in Fig.9, the (P-1) A nodes of PSFG which are connected serially are combined into the pipeline form of (P-2) A nodes. And these pipeline forms of A nodes are constructed by using the pipeline XOR tree. To meet the time requirement there is no need of padding '0' at input due to the AND cell is organized in parallel. The function is as same as the structure-I.

3.4 Structure-III for digit-serial RB multiplier

In this, the bit-addition and bit-multiplication are carried out concurrently and hence the throughput of the desired structure can be increased. The structure-III for digit-serial RB multiplier is shown in Fig.10, which contains (P+1) PPGUs and the each PPGU is with the single AND cell, single XOR cell and two register cells and the first output of this structure-III can be obtained at (P+Q+1) cycles. And at Q cycles the consecutive output is obtained.

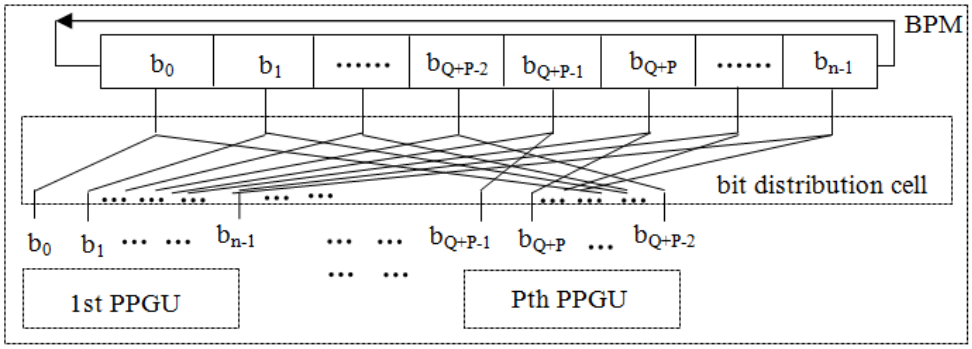


Fig.5. structure of the bit-permutation module (BPM)

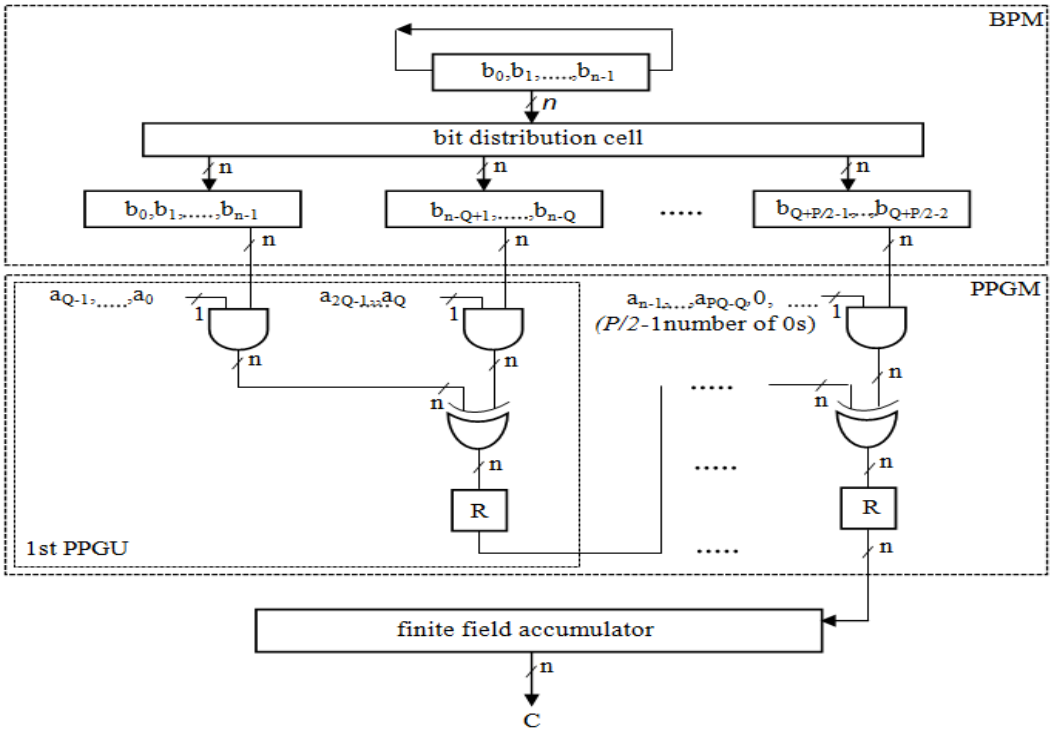


Fig.6. Structure-I for digit-serial RB multiplier when d=2.

The fig.5 shows the structure of the bit-permutation module, and fig.7 (a), 7(b) and fig.7(c) shows the structure of AND cell, XOR cell and register cell of PPGM. Which the inputs are given parallel to the AND cell and obtain the output parallel and also which is done similar to the XOR cell and register cell. This consists of n parallel inputs and n parallel outputs.Fig.8. Shows the structure of finite field accumulator, the finite field accumulator also consists of XOR cell and register cell with the parallel inputs and parallel outputs.

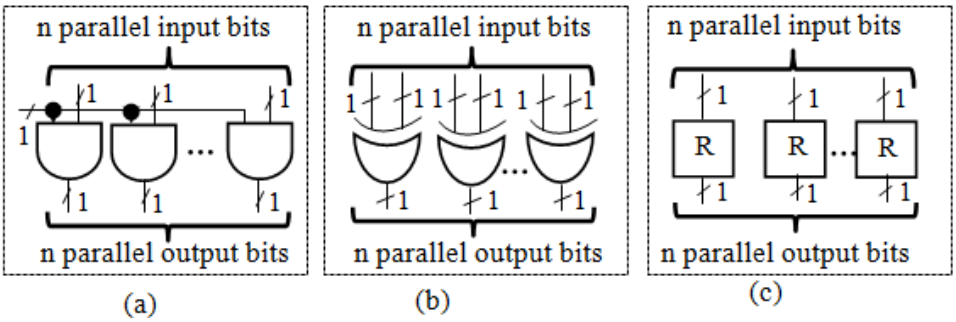


Fig.7 (a) Structure of AND cell in partial product generation module. (b) Structure of XOR cell in partial product generation module. (C) Structure of register cell in partial product generation module.

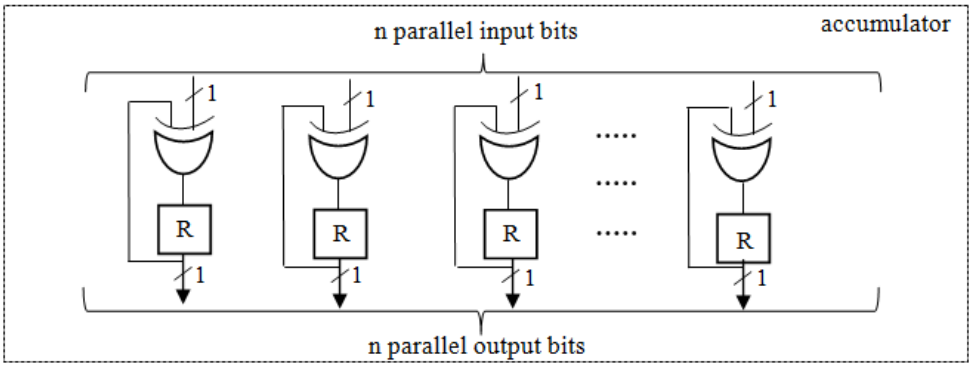


Fig.8. Structure of the finite field accumulator

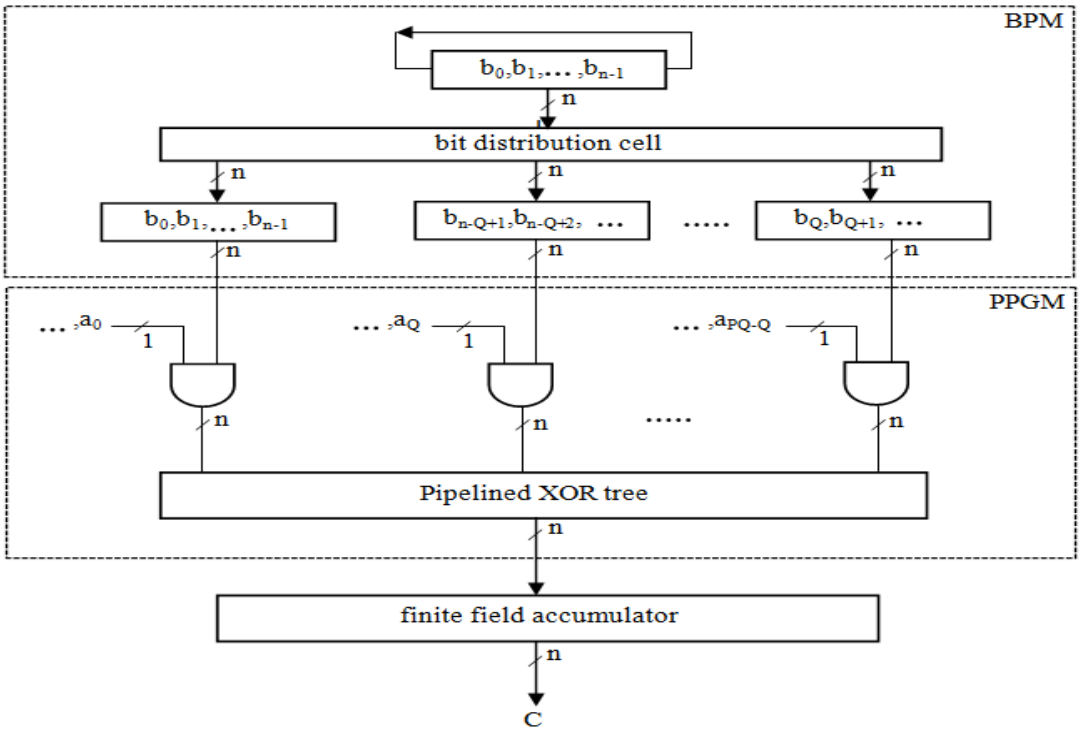


Fig.9. Structure-II for digit-serial RB multiplier

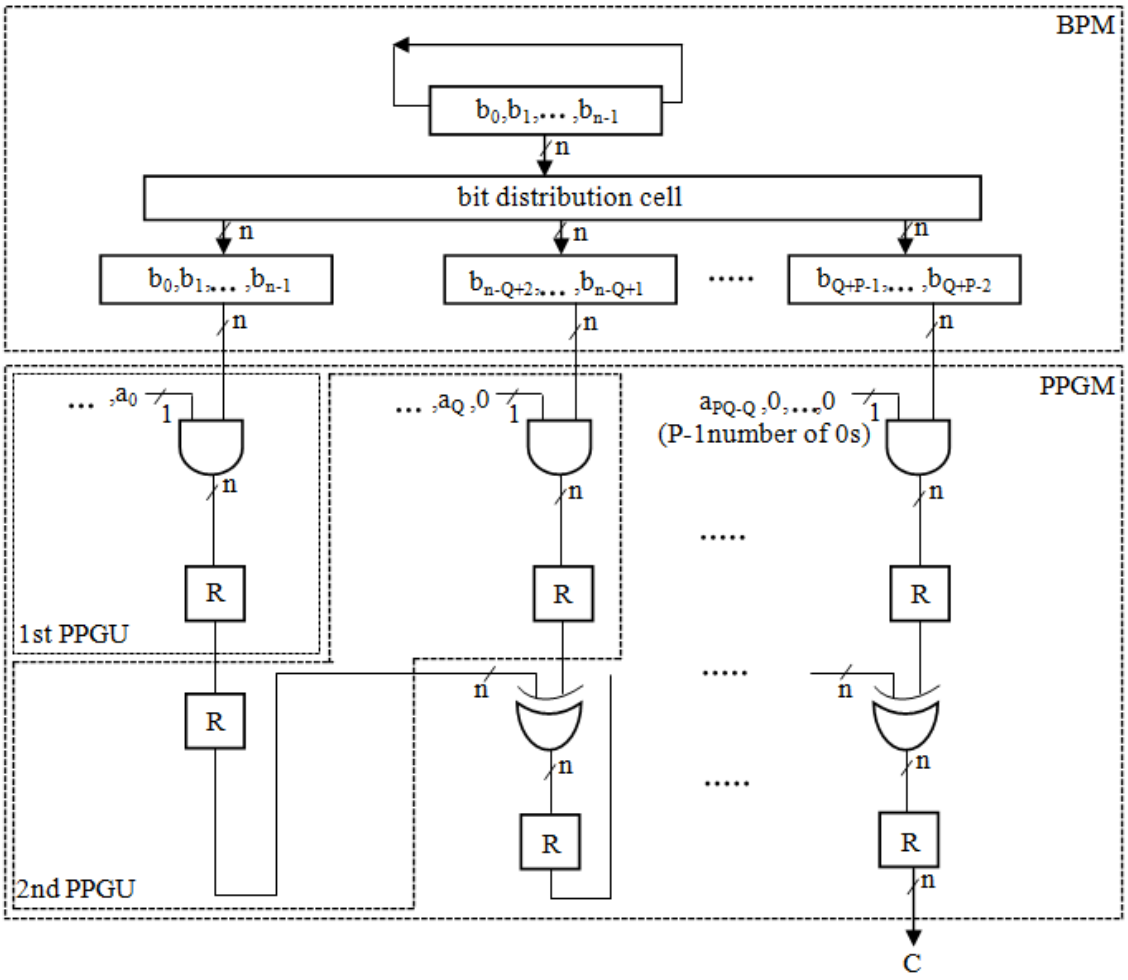


Fig.10. Structure-III for digit-serial RB multiplier

IV. Implementation And Simulation Results

The proposed structures (case 1, case 2, and case 3) are written in a Verilog HDL, synthesized and simulated using Xilinx 12.2.The simulation results and RTL schematic of 10 bit Signal-flow graph(SFG), Processor-space flow-graph (PSFG) and proposed structures (case 1, case 2, and case 3) are shown below.

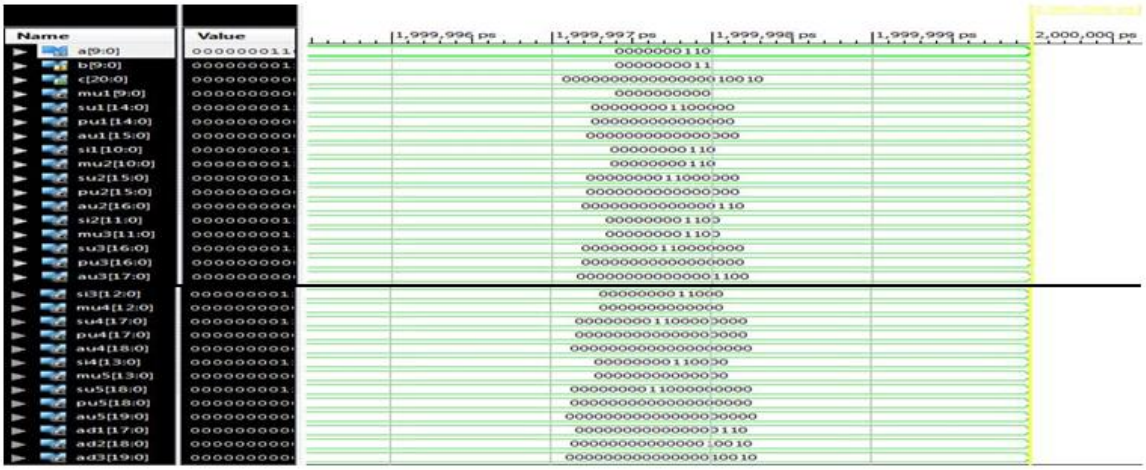


Fig.11. Simulation result of 10 bit Signal-flow graph (SFG)

The simulation result of 10- bit Signal-flow graph (SFG) is shown in Fig.11.The inputs are a=0000000110 and b=0000000011 and the output is c=00000000000000010010 obtained by performing the one

bit shifting, circular bit shifting, addition and multiplication. The detailed view of RTL Schematic of 10 bit Signal-flow graph (SFG) is shown in Fig.12. Here the input operands are10 bit a and b which obtain the 21 bit output c.

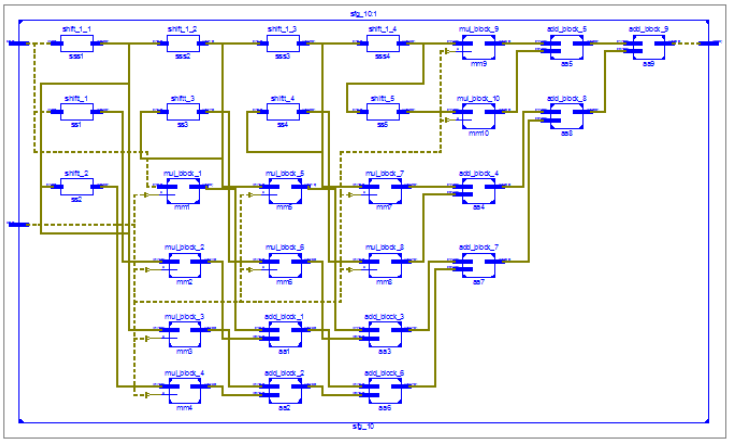


Fig.12. Detailed view of RTL Schematic of 10 bit signal-flow graph (SFG)

The simulation result of 10-bit Processor-space flow-graph (PSFG) is shown in Fig.13.The inputs are a=0000000110 and b=0000001000 and the output obtained is c= 00000000000000011000**.**

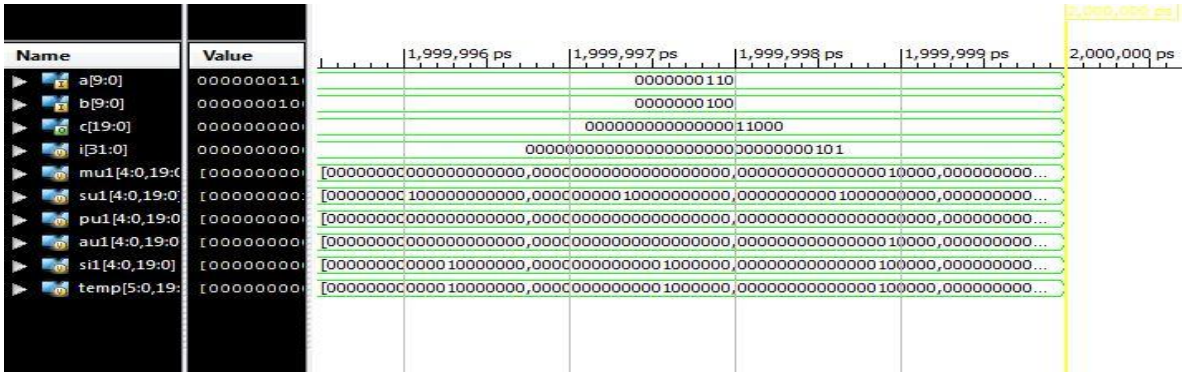


Fig.13. Simulation result of 10 bit PSFG

The detailed view of RTL schematic of 10 bit processor-space flow-graph (PSFG) is shown in Fig.14. Here the input operands are 10-bit a and b and the output obtained is 20 bit c.

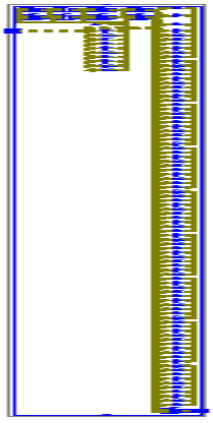


Fig.14. Detailed view of RTL Schematic of 10 bit Processor-space flow-graph (PSFG)

Case 1: The Simulation result of 10 bit structure-I for digit-serial RB multiplier is shown in Fig.15. The inputs are a=00000000100 and b=0000000011 and the output obtained is c=0000001100, by performing the shifting, multiplication and addition operations.

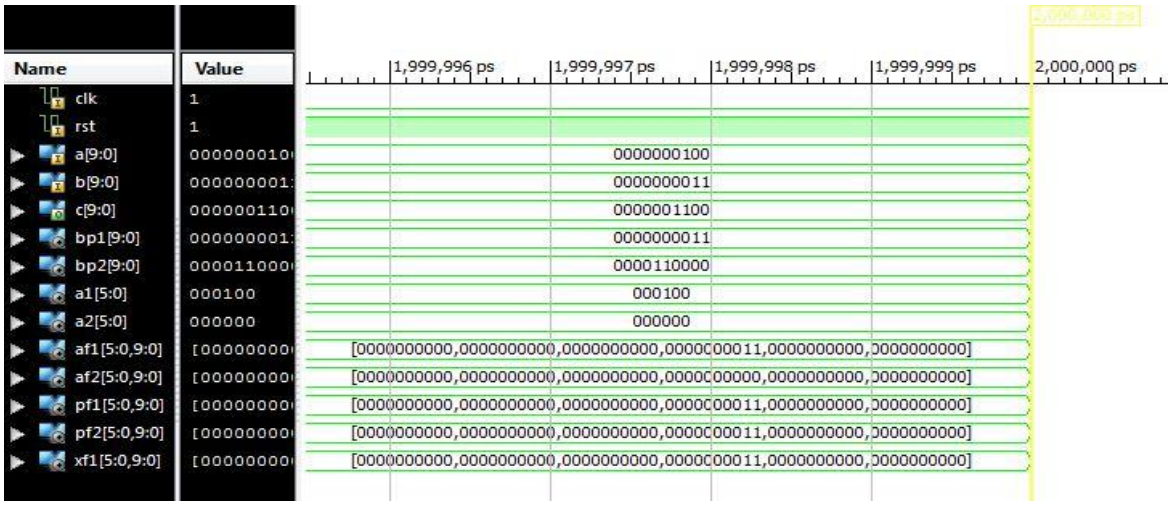


Fig.15. Simulation result of 10 bit Structure-I for digit-serial RB multiplier

The detailed view of RTL schematicof 10 bit Structure-I is shown in Fig.16. Consists of 10-bit input operands a and b with clock and reset, which obtain the 10-bit output c.

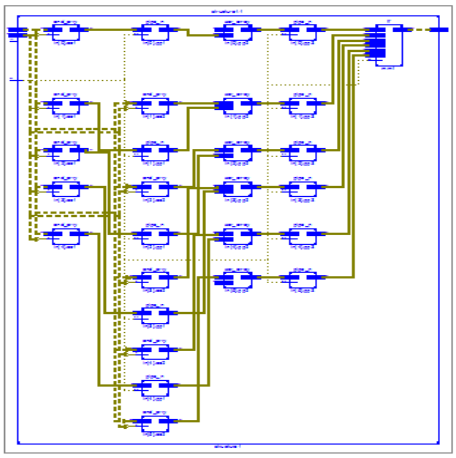


Fig.16. Detailed view of RTL Schematic of 10 bit Structure-I.

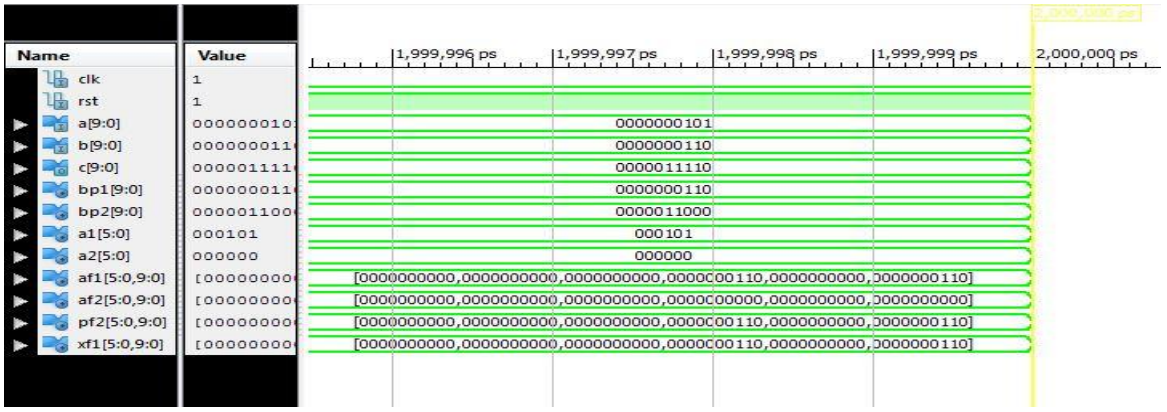


Fig.17. Simulation result of 10 bit structure-I for digit-serial RB multiplier when d=2

The Simulation result of 10 bit structure-I for digit-serial RB multiplier when d=2 is shown in Fig.17.The input operands are a=0000000101 and b=0000000110 with clock=1 and reset=0 and the output obtained is c=0000011110.The detailed view of RTL schematicof 10 bit Structure-I when d=2 is shown in Fig.18.consists of 10-bit a and b operands with clock and reset, which obtain the output 10-bit c.

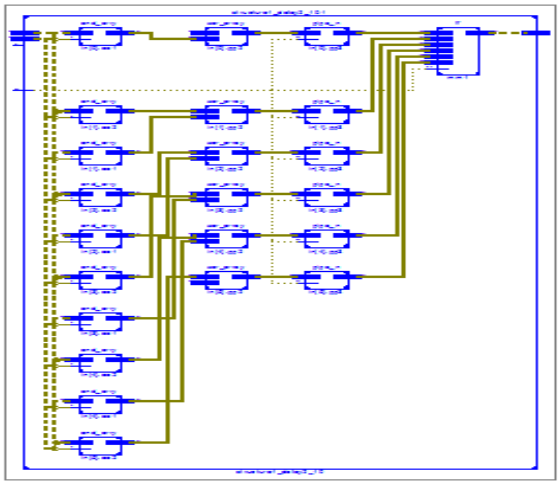


Fig.18.Detailed view of RTL Schematic of 10 bit Structure-I when d=2

Case 2: The Simulation result of 10 bit Structure –II for digit-serial RB multiplier is shown in Fig.19.The input operands are a=0000000011, b=0000000101 with clock=1 and reset=0, and the output obtained is c=0000001111.

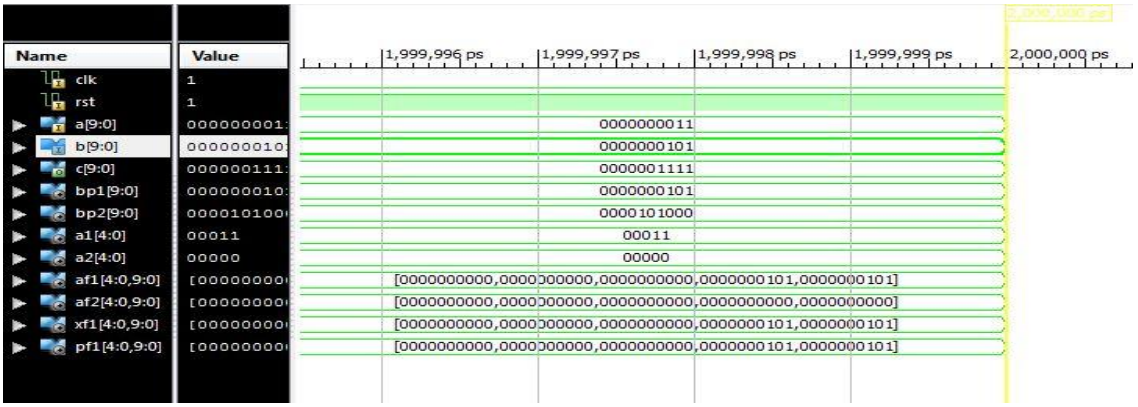


Fig.19 Simulation result of 10 bit Structure-II for digit-serial RB multiplier

The detailed view of RTL schematicof 10 bit Structure-II is shown in Fig.20.consists of 10-bit input operands a and b with clock and reset, and obtained output is 10-bit c.

Case 3: The Simulation result of 10 bit Structure –III for digit-serial RB multiplier is shown in Fig.21.The input operands are a=0000000011,b=0000001000 with clock=1 and reset=0 and which obtain the output c=0000011000.The detailed view of RTL schematic of 10 bit Structure-III is shown in Fig.22. Consists of 10-bit input operands a, b with clock and reset and which obtain the 10-bit output c.

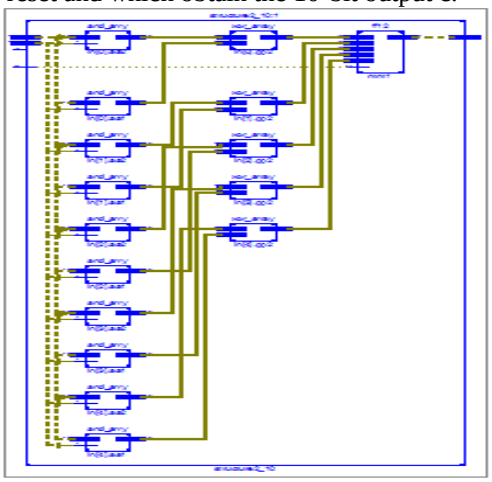


Fig.20 Detailed view of RTL Schematic of 10 bit Structure-II

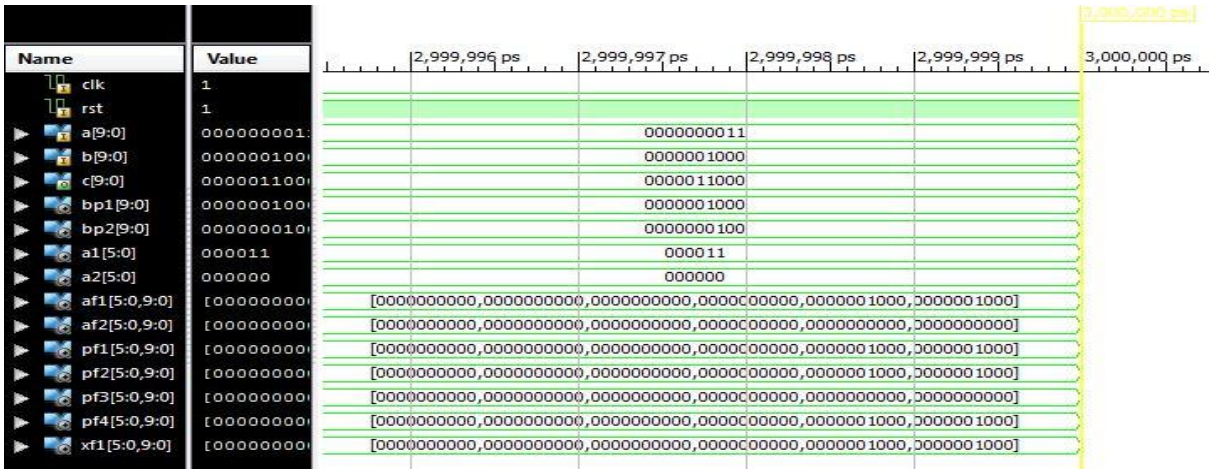


Fig.21. Simulation result of 10 bit Structure-III for digit-serial RB multiplier

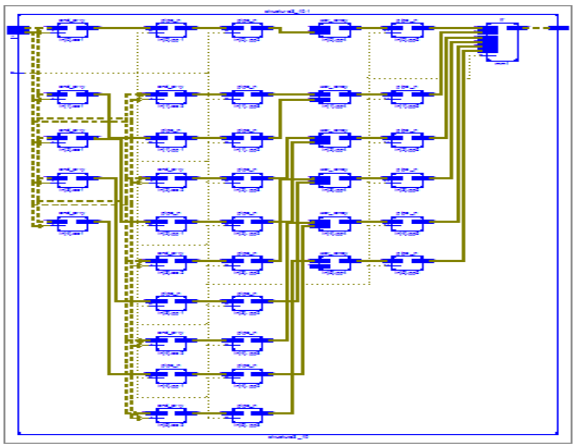


Fig.22. Detailed view of RTL Schematic of 10 bit Structure-III

V. Conclusion

The proposed structures (Case 1, Case 2, and Case 3) for digit-serial RB multipliers are implemented in Verilog HDL by using novel recursive decomposition algorithm. The synthesis is done for 10 bit proposed structures (Case 1, Case 2, and Case 3) and is simulated by using Xilinx 12.2.The proposed structures are implemented to obtain high-throughput, by projection of signal-flow graph to the processor-space flow-graph. And these multipliers are used based on application requirement mostly in cryptographic applications. The detailed RTL schematic of proposed structures is also obtained.

References

[1]. H.Niederrieter,*Introduction to finite fields and their applications* 2nd edition (Cambridge,UK:Cambridge University Press, 1997).
[2]. Swamy.M.N, Cryptography applications of bhaskara equations,*IEEE Trans.Circ.Sys.I, vol.54 (7)*, 2007, 927-928.
[3]. Retheesha.D and Ajitha.S.S, Efficient implementation of bit parallel finite field multipliers, *IJRET,vol 3*, 2014, 661-667.
[4]. Jun-CheolJeon and Kee-W.Kim, Finite field arithmetic architecture based on cellular array,*International Journal of Cyber-Security Forensis, 1(2)*, 2012,122-129.
[5]. L.S.Hsu, Comparison of VLSI architecture of finite field multipliers using dual, normal or standard basis,*IEEE Trans.Comp.*, 1987,63-75.
[6]. Zhi-Hong Mao and J.Xie,High-throughput finite field multipliers using redundant basis for FPGA and ASIC implementation, *IEEE Trans. Circ. Sys-I, 62(1)*, 2015, 110-119.
[7]. G.Yuvaraj, Design of word level multiplication algorithm on reordered normal basis, *International Journal of innovations in engineering and* technology,*1(4)*, 2012, 88-94.
[8]. Peter.K and A.S.Nielsen, Redundant radix representation of rings, *IEEE Trans. Comp., 48(11)* ,1999, 1153-1165.
[9]. Yasser Salem and RosliSalleh, A bit-serial multiplier architecture for finite fields over Galois fields, *Journal of Computer Science, 6(11)*, 2010, 1237-1246.