IJSEAT International Journal of Science Engineering and Advance Technology

Design And Implementation Of Efficient Multiplier Using Vedic Sutras

Theppala Devi*[1], Perabathula Satyavani[2]
M.Tech Scholar, Department of ECE.[1]
Assist.Prof, Department of ECE, Kakinada Institute of Engineering &Technology
East Godavari Dist., AP, India.[2]

***Abstract*— The primary target of this venture is to outline a productive excess parallel multiplier utilizing recursive disintegration calculation. By reasonable projection of SFG of proposed calculation and distinguishing appropriate cut-sets for bolster forward cut-set retiming, three novel high-throughput digit-serial RB multipliers are inferred to accomplish altogether less zone time-control complexities than the current ones. It is demonstrated that the proposed high-throughput structures are the best among the relating plans, for FPGA and ASIC execution. This undertaking is actualized by utilizing vedic duplications. Urdhva Tiryakbhyam sutra is utilized for planning this multiplier. Range is fundamental diminishment with this kind of multiplier. Through productive projection of flag stream chart (SFG) of the proposed calculation, a profoundly normal processor-space stream diagram (PSFG) is inferred.

Catchphrases: Field Programmable Gate Arrey (FPGA), Application Specific IC (ASIC), processor-space stream (PSFG), Finite Impulse Response(FIR).

I. INTRODUCTION

Limited Field augmentation over Galois Field ( ) is an essential operation every now and again experienced inmodern cryptographic frameworks, for example, the elliptic bend cryptography (ECC) and blunder control coding [1]– [3]. Also, augmentation over a limited field can be utilized further to perform other field operations, e.g., division, exponentiation, and reversal [4]– [6]. Augmentation over can be executed on a broadly useful machine, however it is costly to utilize a universally useful machine to actualize cryptographic frameworks in cost-delicate buyer items. In addition, a low-end chip can't meet the constant necessity of various applications since word-length of these processors is too little contrasted and the request of regular limited fields utilized as a part of cryptographic frameworks. The greater part of the ongoing applications, along these lines, require equipment execution of limited field number-crunching operations for the advantages like ease and high-throughput rate. The decision of premise to speak to handle components, to be specific the polynomial premise, typical premise, triangular premise and excess premise (RB) majorly affects the execution of the number juggling circuits [7]–[9]. The multipliers in view of RB [6], [10] have increased huge consideration as of late because of their few focal points. Not exclusively do they offer free squaring, as typical premise does, yet additionally include bring down computational intricacy and can be actualized in very standard figuring structures [10]– [14]. There are diverse sorts of bases to speak to handle components, those are polynomial premise, typical premise, triangular premise and RB, and the decision of portrayal of field components majorly affects the execution of the math circuits [7]– [9], [4]. A few calculations for essential number juggling operations in GF (2) are appropriate for both equipment and programming usage have been as of late created. In light of a few preferences of the RB based multipliers [6], [10] they have increased noteworthy consideration as of late. Like typical premise multipliers, RB multipliers offer free squaring, they additionally include bring down computational multifaceted nature and can be executed in profoundly normal registering structures [10]– [14]. A few digit-level serial/parallel structures for RB multiplier over GF (2?) have been accounted for over the most recent couple of years [1]– [1]. A productive serial/parallel multiplier utilizing repetitive portrayal has been exhibited [1].

II. REDUNDANT BASIS REPRESENTATION:

A repetitive paired portrayal (RBR) is a numeral framework that utilizations a bigger number of bits than expected to speak to a solitary parallel digit with the goal that most numbers have a few portrayals. A RBR is not at all like common paired numeral frameworks, including two's supplement, which utilize a solitary piece for every digit. A large number of a RBR's properties contrast from those of normal paired portrayal frameworks. Above all, a RBR permits expansion without utilizing a common carry.[1] When contrasted with non-excess portrayal, a RBR makes bitwise sensible operation slower, yet math operations are quicker when a more noteworthy piece width is used.[2] Usually, every digit has its own particular sign that is not really the same as the indication of the number spoke to. At the point when digits have signs, that RBR is additionally a marked digit portrayal.

A RBR is a place-esteem documentation framework. In a RBR, digits are sets of bits, that is, for each place, a RBR utilizes a couple of bits. The esteem spoke to by a repetitive digit can be discovered utilizing an interpretation table. This table shows the numerical estimation of every conceivable combine of bits.

Digit Serial RB Multiplier:

In Digit serial RB multiplier [12],[16] digit insightful incomplete items for the digit serial augmentation where operands An and B are disintegrated into various digits, and afterward the expansion of those fractional items is done to process the item word.

Expecting x to be a primitive nth foundation of solidarity, components in GF (2m) can be spoken to in the frame:

A= ?0 + ?1x + … . + ??−1??−1 - 1

Where ?? ϵ GF (2), for 0≤ i ≤ n-1, with the end goal that the set {1, x,?2,… .., ??−1} is characterized as the RB for GF (2?) components, where n is a positive whole number at the very least m [6], [10].

For a GF (2m), when (m+1) is prime and 2 is a primitive root modulo (m+1), there exists a sort I ideal typical premise (ONB) [10], where x is a component of GF (2m), and n=m+1**.**

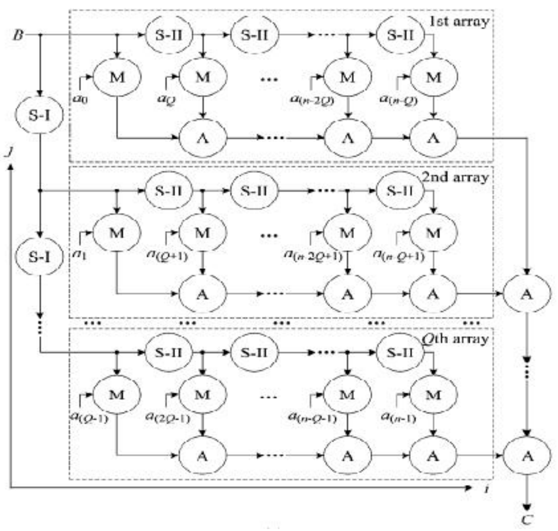


Fig1: Signal-flow graph (SFG) for parallel realization of RB multiplication

Let A, B ϵ GF (2) be expressed in RB representation as

A= $aixin-1i=0$ (2)

B= $bixin-1i=0$ (3)

where ai, bi ϵ GF (2). Let C be the product of A and B, which can be expressed as follows

C= A·B = $(xibi)n-1i=0$·A

= $(n-1i=0\ bix(i+j))ajn-1j=0$

= $(n-1j=0\ b(i-j)nxi)ajn-1i=0$

= $(n-1i=0\ b(i-j)n\ aj)xin-1j=0$ (4)

where (i-j)n denotes modulo n reduction. Define C= $cixin-1i=0$ where ci ϵ GF (2), then

$ci = b\ i-j\ n\ ajn-1i=0$ (5)

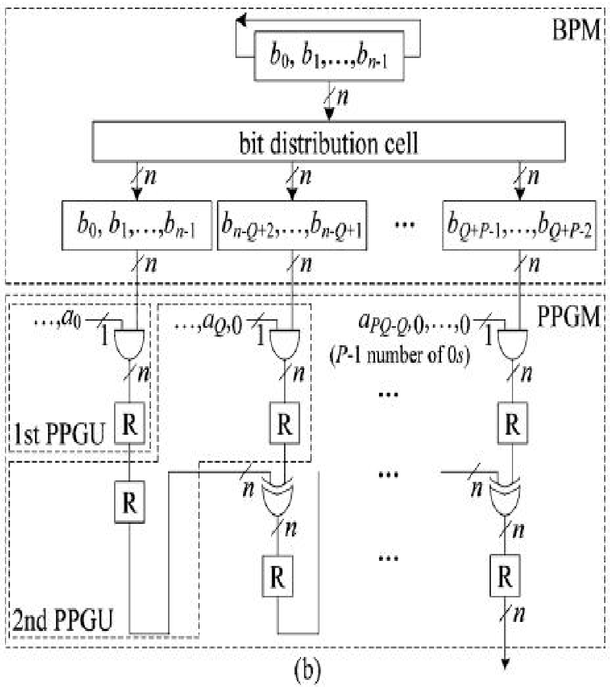


Fig2: Modified RB Structure

In Digit serial RB Multiplier in a bit level matrix vector form [16], both the operands are decomposed into a number of blocks to achieve digit serial multiplication and after that the partial products corresponding to these blocks are added together to obtain desired product word. Considering equations (1) to (4) of Digit Serial RB Multiplier (III.A)
Where (i-j)n denotes modulo n reduction. Define C in a bit level matrix vector form as:
c0c1⋮cn−1 = b0b1 bn−1 b0⋯b1b2⋮ ⋮⋱⋮bn−1bn−2⋯b0 a0a1⋮an−1 (5)
Looking at the structure of bit matrix in (5), n bit shifted (reduced) forms of operand B can be defined as follows
$B0 = bi0xin-1i=0 = b0 + b1x + …. + bn-1xn-1$
$B1 = bi1xin-1i=0 = bn-1 + b0x + …. + bn-2xn-1$ (6)
………………………………………………
$Bn-1 = bin-1xin-1i=0 = b1 + b2x + …. + b0xn-1$
where
$b0i+1 = bn-1i$
$bji+1 = bj-1i$; for 1 ≤ j ≤ n-2 (7)
The recursions on (7) can be extended further to have
$bji+s$ = bn−s+jifor 0≤j≤s−1bj−si otherwise (8)
where 1 ≤ s ≤ n-1.
Let Q and P be two integers such that n = QP + r, where 0 ≤ r < P. for simplicity of discussion, it is assumed that r=0, and decompose the input operand A into Q number of bit vectors Au for u = 0,1,….., Q-1, as follows:
$A0 = [a0\ aQ\ ….\ an-Q]$
$A1 = [a1\ aQ+1\ ….\ an-Q+1]$
………………………. (9)
$AQ-1 = [aQ-1\ a2Q-1\ ….\ an-1]$
Similarly, Q number of shifted operand vector Bu can be generated, for u = 0,1,…., Q-1
$B0 = [B0\ BQ\ ….\ Bn-Q]$
$B1 = [B1\ BQ+1\ ….\ Bn-Q+1]$
………………………. (10)
$BQ-1 = [BQ-1\ B2Q-1\ ….\ Bn-1]$
The product C=A·B given by the *bit level matrix* vector product in (5) can be decomposed into Q inner products of vectors Au and Bu for u = 0,1,…., Q-1 as:
C=A·B = $B0A0T + B1A1T + …. + BQ-1AQ-1T$
C = $BuQ-1u=0AuT = Cu\ Q-1u=0$ (11)
Where $Cu = BuAuT$

Note that each Au for u = 0,1,….,Q-1 is a P point bit vector and each Bu for u = 0,1,….,Q-1 is a P point bit shifted forms of operand B. From (11) and (12) it can find that the desired multiplication can be performed by Q cycles of successive accumulation of Cu for u = 0,1….,Q-1 while each Cu can be computed as $Cu = Bu+vQau+vQP-1v=0$.

The partial products generated in this multiplier are lesser in numbers than those generated in above multipliers, which reduces the computation time, and also reduces register and adder complexities of RB multipliers.

The partial products generated in this multiplier are lesser in numbers than those generated in above multipliers, which reduces the computation time, and also reduces register and adder complexities of RB multipliers.

I. VEDIC SUTRAS:

The "Vedic Mathematics" is called so as a result of its cause from Vedas. To be more particular, it has started from "Atharva Vedas" the fourth Veda. "Atharva Veda" manages the branches like Engineering, Mathematics, model, Medicine, and every single other science with which we are today mindful of. The Sanskrit word Veda is gotten from the root Vid, which means to know unbounded. The word Veda covers all Veda-Sakhas known to mankind. The Veda is an archive of all information, fathomless, regularly uncovering as it is dove deeper.Vedic science, which rearranges number juggling and arithmetical operations, has progressively discovered acknowledgment the world over. Specialists propose that it could be a helpful instrument for the individuals who need to take care of numerical issues speedier by the day. It is an old system, which disentangles augmentation, detachability, complex numbers, squaring, cubing, square roots and 3D square roots.

URDHVA TIRYAGBHYAM

Urdhva – Triyakbhyam is the general equation appropriate to all instances of duplication and furthermore in the division of a huge number by another extensive number. It implies vertically and across. We examine duplication of two, 4 digit numbers with this strategy [8-9]. Ex.1. the result of 1111 and 1111 utilizing Triyakbhyam (vertically and across) is given below.Methodology of Parallel Calculation1 1 `1 1 1 1 1 1 1 x 1 = 1 1 1 1 1 1 1 1 1 1 x 1+1 x 1 = 2
1 1 1 1 1 1 1 1 1 x 1+1 x 1+ 1 x 1 = 3
1 1 1 1 1 1 1 1 1 x 1+1 x 1+1 x 1+1 x 1 = 4
1 1 1 1 1 1 1 1 1 x 1+1 x 1+ 1 x 1 = 3
1 1 1 11 1 1 1 1 x 1+1 x 1 = 2
1 1 1 11 1 1 1 1 x 1 = 1
Final answer = 1 2 3 4 3 2 1

II. MULTIPLIER ARCHITECTURE

The equipment design of 2X2, 4x4 and 8x8 piece Vedic multiplier module are shown in the beneath segments. Here, "Urdhva-Tiryagbhyam" (Vertically and Crosswise) sutra is utilized to propose such engineering for the duplication of two parallel numbers. The magnificence of Vedic multiplier is that here fractional item era and increases are done simultaneously. Subsequently, it is very much adjusted to parallel preparing. The component makes it more appealing for parallel increases. This thusly decreases delay, which is the essential inspiration driving this work.A. Vedic Multiplier for 2x2 piece Module The technique is clarified beneath for two, 2 bit numbers An and B where A = a1a0 and B = b1b0. Right off the bat, the minimum huge bits are duplicated which gives the slightest huge piece of the last item (vertical). At that point, the LSB of the multiplicand is increased with the following higher piece of the multiplier and included with, the result of LSB of multiplier and next higher piece of the multiplicand (transversely). The whole gives second piece of the last item and the convey is included with the fractional item acquired by increasing the most noteworthy bits to give the aggregate and convey. The whole is the third relating bit and convey turns into the fourth piece Of the finel item. The 2X2 Vedic multiplier module is actualized utilizing four info AND doors and two half-adders which is shown in its piece. It is discovered that the equipment engineering of 2x2 piece Vedic multiplier is same as the equipment design of 2x2 piece regular Array Multiplier [2]. Subsequently it is presumed that increase of 2 bit paired numbers by Vedic technique does not made noteworthy impact in change of the multiplier"s productivity. Precisely we can express that the aggregate deferral is just 2-half snake delays, after definite piece items are created, which is fundamentally the same as Array multiplier. So we change over to the execution of 4x4 piece Vedic multiplier which utilizes the 2x2 piece multiplier as an essential building square. A similar technique can be stretched out for input bits 4 and 8. Yet, for higher no. of bits in input, little change is required

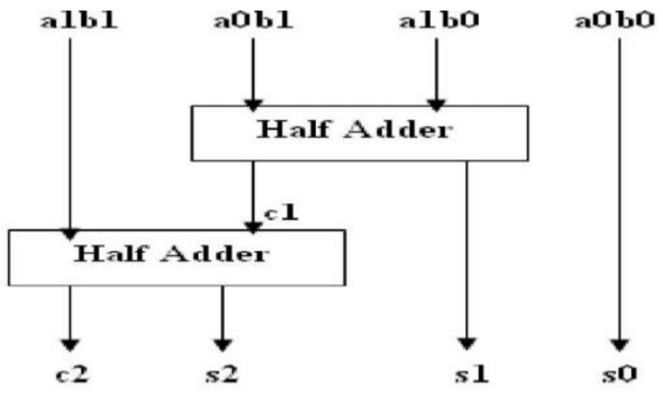


Figure 3: Block Diagram of 2x2 bit Vedic Multiplier

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules as discussed in Fig. 3.Let"s analyze 4x4 multiplications, say A= A 3 A2 A1 A0 and B= B3 B2 B1 B0. The output line for the multiplication result is – S 7 S 6 S 5 S 4 S 3 S 2 S 1 S 0 .Let"s divide A and B into two parts, say A3 A2 & A1 A0 for A and B3 B2 & B1B0 for B. Using the fundamental of Vedic multiplication, taking two bit at a timeand using 2 bit multiplier block, we can have the following structure for multiplication as shown in below figure

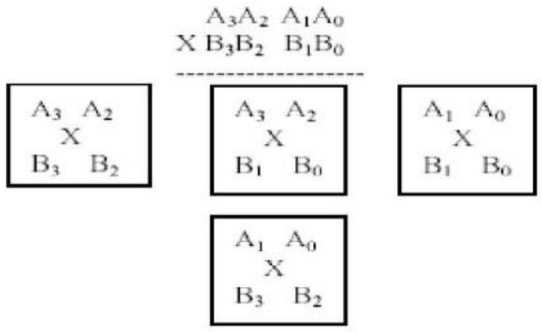


Figure 4: Sample Presentation for 4x4 bit Vedic Multiplication

Sample Presentation for 4x4 bit Vedic Multiplication Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A 1A0 and B 1B0. The last block is 2x2 bit multiplier with inputs A 3 A2 and B 3 B2. The middle one shows two 2x2 bit multiplier with inputs A3 A2 & B1B0 and A 1A0 & B3 B2. So the final result of multiplication, which is of 8 bit, S7 S 6 S 5 S 4 S 3 S 2 S1 S0.. To get final product (S7 S 6 S 5 S 4 S 3 S 2 S 1 S 0), four 2x2 bit Vedic multiplier and three 4-bit Ripple-Carry (RC) Adders are required. The proposed Vedic multiplier can be used to reduce delay. Early literature speaks about Vedic multipliers based on array multiplier structures. On the other hand, we proposed a new architecture, which is efficient in terms of speed. , helps us to reduce delay. Interestingly, 8x8 Vedic multiplier modules are implemented easily by using four 4x4 multiplier modules.

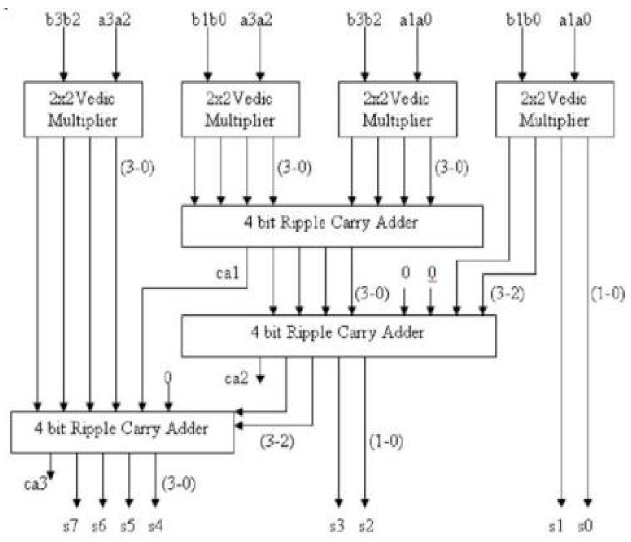


Figure 5: Block Diagram of 4x4 bit Vedic Multiplier

Vedic Multiplier for 8x8 bit Module The 8x8 bit Vedic multiplier module as shown in the above block diagram can be easily implemented by using four 4x4 bit Vedic multiplier modules as discussed in the previous section Let‟s analyze 8x8 multiplications, say A= A7 A6 A5 A4 A3 A2 A1 A0 and B= B 7 B6 B5B4 B3 B2 B1B0. The output line for the multiplication result will be of 16 bits as – S 15 S 14 S 13 S 12 S 11 S 10 S 9 S 8 S 7 S 6b S 5 S 4 S 3 S 2 S 1 S 0. Let‟s divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL. The 16 bit product can be written as: Using the fundamental of Vedic multiplication, taking four bits at a time and using 4 bit multiplier block as discussed we can perform the multiplication. The outputs of 4x4 bit multipliers are added accordingly to obtain the final product. Here total three 8 bit Ripple-Carry Adders are required.

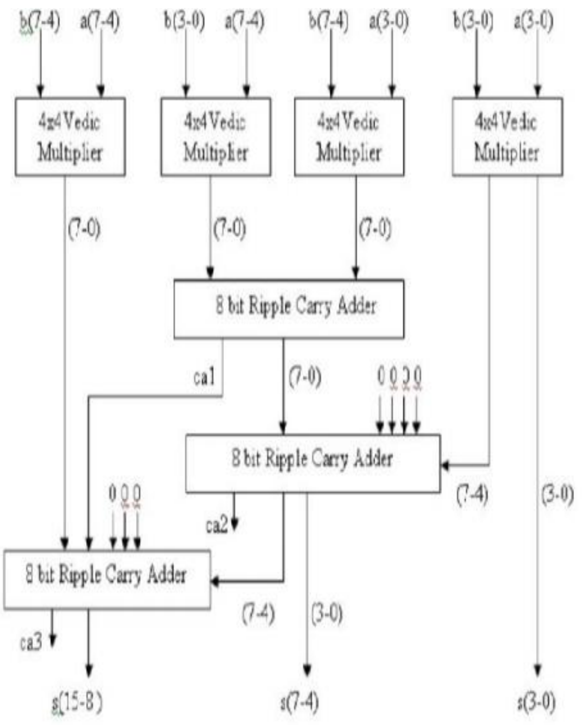


Figure 6: Block Diagram of 8x8 bit Vedic Multiplier

Generalized Algorithm for N x N bit Vedic Multiplier We can generalize the method as discussed in the previous sections for any number of bits in input. Let, the multiplication of two N-bit binary numbers (where N = 1, 2, 3…N, must be in the form of 2 N) A and B where A = AN.....A3 A2 A1 and B = BN….B3 B2 B1. The final multiplication result will be of (N + N) bits as S = S(N + N) ....S3 S S Step 1: Divide the multiplicand A and multiplier B into two equal parts, each consisting of [N to (N/2)+1] bits and [N/2 to 1] bits respectively, where first part indicates the MSB and other represents LSB. Step 2: Represent the parts of A as AM and A L, and parts of B as B M and B L. Now represent A and B as A M AL and B M BL respectively. Step 3: For A X B, we have general format as shown in below figure.

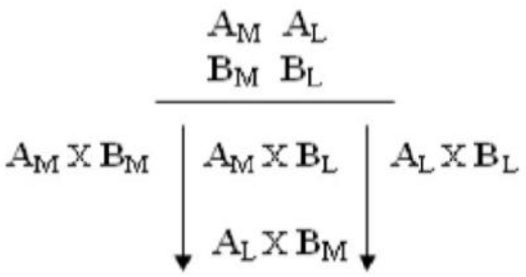
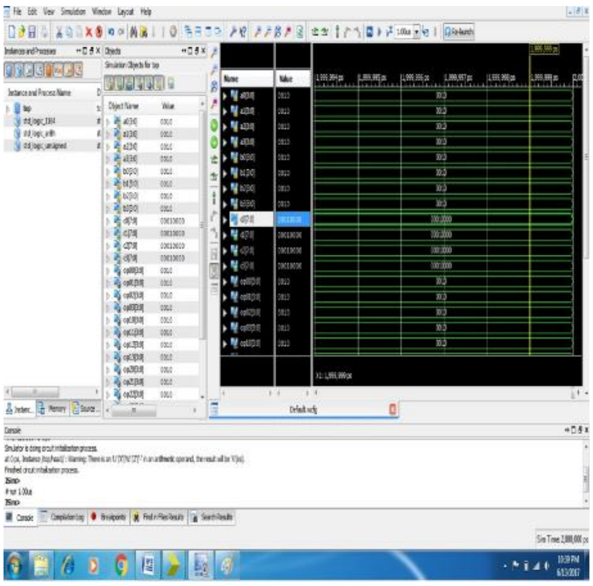


Figure 7: General Representation for vedic multiplication

III. RESULTS



```
Simulator is doing circuit initialization process.
at 0 ps, Instance /top/hadd/ : Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
Finished circuit initialization process.
ISim>
# run 1.00us
ISim>
```

Figure 8: Simulation Results

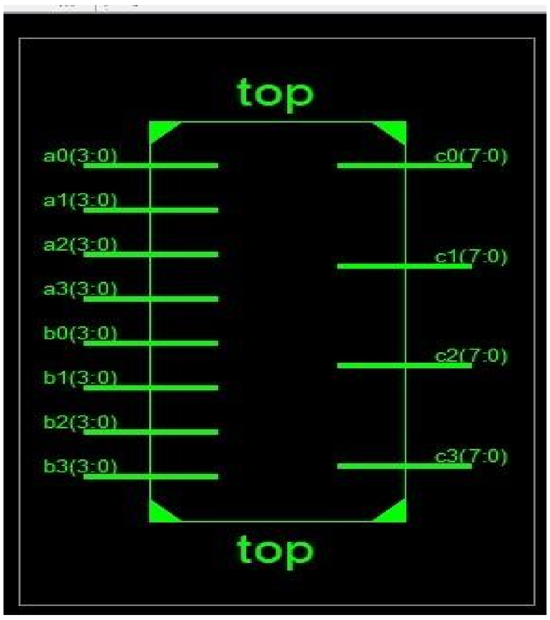


Figure 9: RTL Schematic

IV. CONCLUSION

RB multipliers over GF (2) are very popular in Elliptic Curve Cryptography because of their negligible hardware cost for squaring and modular reduction. Word Level RB multiplier is the most efficient among all multipliers in terms of hardware utilization. Digit serial RB multiplication in a bit level matrix vector form is most efficient in terms of area-time complexities. Future works can be done to find out new methods to obtain partial products in lesser time and with less hardware requirements using vedic sutra.

V. REFERENCES

[1] Alessandro Cilardo, Luigi Coppolino, Nicola Mazzocca, and Luigi Romano, "Elliptic Curve Cryptography Engineering," proc. of IEEE, vol.94, no.2, pp.395-406, Feb.2006.

[2]N.R.Murthy and M.N.S.Swamy, "Cryptographic applications of brahmagupta-bhaskara equation," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.53, no.7, pp.1565-1571, 2006.

[3]L.Song and K.K.Parhi, "Low-energy digit-serial/parallel finite field multipliers," J.VLSI Digit. Process, vol.19, pp.149-166, 1998.

[4]P.K.Meher, "On efficient implementation of accumulation in finite field over GF (2^m) and its applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.17, no.4, pp.541-550, 2009.

[5]L.Song, K.K.Parhi, I.Kuroda, and T.Nishitani, "Hardware / software codesign of finite field data path for low-energy Reed-Solomn codecs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.8, no.2, pp.160-172, Apr.2000.

[6]G.Drolet, "A new representation of elements of finite fields GF (2^m) yielding small complexity arithmetic circuits," IEEE Trans. Comput., vol.47, no.9, pp.938-946, 1998.

[7]C.Y.Lee, J.S.Horng, I.C.Jou, and E.H.Lu, "Low-complexity bit parallel systolic Montgomery multipliers for special classes of GF (2^m)," IEEE Trans. Comput., vol.54, no.9, pp.1061-1070, Sep.2005.

[8]P.K.Meher, "Systolic and super-systolic multipliers for finite field GF (2^m) based on irreducible trinomials," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.55, no.4, pp.1031-1040, May 2008.

Author's profile

Ms. Theppala Devi Has Completed Her B.Tech In E.C.E Department From Jntu College Of Engineering Vizianagaram Presently She is Pursuing Her Masters in VLSI Design in Kakinada Institute of Engineering and Technology

Mrs. P. satyavani Has Completed Her M.Tech (Embedded Systems) From R.R.S College of Engineering And Technology She is Having 4 Years Of Experience In Academics, Currently Working as Assistant Professor at Kakinada Institute Of Engineering and Technology