IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

Polynomial Based Mastrovito Multipliers for FFT Applications for Secure Data Encryption and Decryption

Swarnim Baghel
M. Tech Scholar
Department of Electronics and Communication
Gyan Ganga Institute of Technology and Science,
Jabalpur (M.P.), INDIA
RGPV University, Bhopal (M.P.)
swarnim.baghel@gmail.com

Sunil Shah
Asst. Professor
Department of Electronics and Communication
Gyan Ganga Institute of Technology and Science,
Jabalpur (M.P.), INDIA
RGPV University, Bhopal (M.P.)
sunilshah@ggits.org

Abstract- **At present scenario Polynomial groundwork multipliers re used for the reason that they're fairly simple to design, and offer scalability for the fields of bigger orders. It's used in Cryptographic and FFT purposes for at ease knowledge encryption and decryption which deals with discrete constitution and mathematical arithmetic. In view that it makes use of modular arithmetic operation, it's observed that it has the latency of m cycles. To beat this problem, an effective low latency polynomial multiplier for speedy Fourier transform (FFT) algorithm which is based on Mastrovito structure has been developed. This multiplier makes use of the notion of parallel processing in which multiplication is decomposed into number of impartial models and "pre-computed addition" systems. Our design has been implemented in VHDL, simulated and synthesized utilizing the Xilinx ISE Design Suite 13.2 device for supply voltage levels from 1.2V to 2.5 V. The multiplier is analyzed in terms of pace, subject overhead and reminiscence. The design includes significantly less prolong and field overhead complexities than the present structure.**

Keywords: **FFT, Mastrovito, VHDL, Polynomial basis.**

I. INTRODUCTION

on this work, our goal is to build a scalable Bernard Law Montgomery modular multiplier centered on irreducible polynomial for speedy Fourier become (FFT) functions. The FPGA code is written in very excessive velocity hardware description language (VHDL) after which synthesized and carried out using Xilinx FPGA contraptions, and we when compared the outcome of Modified booth situated modular multiplier with array, Wallace tree and sales space multiplier.

A. CRYPTOGRAPHY /FFT APPLICATIONS

Cryptography is the artwork of designing and breaking ciphers. Almost always, secret–key. Cryptography used to be utilized by the army and diplomatic services for providing secure verbal exchange, where two speaking events share a secret key that will have to be dispensed in some relaxed means. Progress of cell internet instruments increased the need for fast Fourier turn into techniques for privacy and authentication of digital data. The invention of public–key cryptography, which assigns two keys (one public and one exclusive) to each and every person, offered approaches for key distribution as well as signing and authenticating digital information. In view that of its complexity, public–key cryptography is usually used for digital signatures and the management of secret keys between two aspects. The encryption of bulk data is as a rule headquartered with secret–key cryptosystems, whereas the secret keys to be shared for a pair of users are dispensed by means of public–key cryptosystems. For a standard public–key cryptosystem to be viewed as "secure", the key length must be about thousand bits or more. [1, 2, 3] .In public–key cryptography, input and output numbers are selected from finite aspect fields. All fast Fourier grow to be operations are made in these finite fields, which map to modular multiplication and modular exponentiations within the digital world. Growing demand for modular multiplication requires rapid modular multiplication algorithms comparable to CSE and modified sales space founded modular multiplication, in order to be described absolutely in this paper.

In the context of speedy Fourier develop into algorithms, a butterfly is a component to the computation that mixes the results of smaller discrete Fourier transforms (DFTs) right into a greater DFT, or vice versa (breaking a larger DFT up into sub transforms). The name "butterfly" comes from the form of the info-float diagram in the radix-2 case, as described below:

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

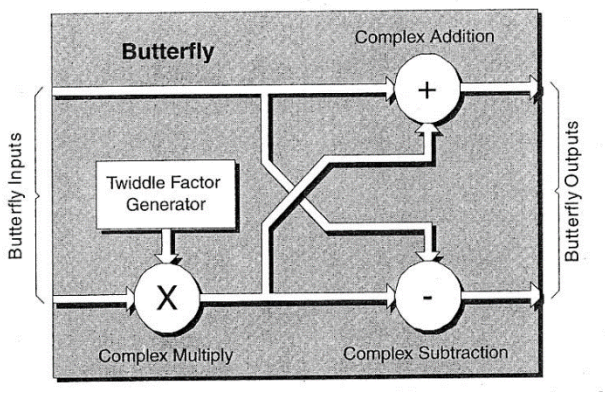

Figure 1: Architecture of 2 point-FFT

Program substitution:
product_1 <= (y_r*w_r) - (y_i*w_i);
product_2 <= (y_r*w_i) + (y_i*w_r);
A_r <= x_r + product_1(7 downto 0);
A_i <= x_i + product_2 (7 downto 0);
B_r <= x_r - product_1 (7 downto 0);
B_i <= x_i - product_2 (7 downto 0);
Substitute '1' for all the values:
Product_1<= (1*1)-(1*1) ;=> 0
Product_2<= (1*1) + (1*1) ;=> 2
A_r<=1+0 ;=> 1
A_i<=1+2 ;=> 3
B_r<=1-0 ;=> 1
B_i<=1-2=> -1
(Compare the values with the simulation window)
Method and diagram shown above is the base architecture of the butterfly structure.

II. Related Work

Software substitution:
product_1 <= (y_r*w_r) - (y_i*w_i);
product_2 <= (y_r*w_i) + (y_i*w_r);
A_r <= x_r + product_1(7 downto 0);
A_i <= x_i + product_2 (7 downto zero);
B_r <= x_r - product_1 (7 downto 0);
B_i <= x_i - product_2 (7 downto zero);
alternative '1' for all the values:
Product_1<= (1*1)-(1*1) ;=> 0
Product_2<= (1*1) + (1*1) ;=> 2
A_r<=1+0 ;=> 1
A_i<=1+2 ;=> 3
B_r<=1-0 ;=> 1
B_i<=1-2=> -1
(examine the values with the simulation window)
approach and diagram proven above is the bottom structure of the butterfly structure.

III. Related Work

In this work, we take into account detection of blunders in polynomial, twin, and normal bases arithmetic operations. Error detection is carried out by way of re computing with the shifted operand process, while the operation unit is in use. This scheme is efficient for pipelined architectures, primarily systolic arrays. Additionally, one semi systolic multiplier for every of the polynomial, dual, style I, and kind II optimal traditional bases is offered [7].
Finite fields were used for countless functions together with error-control coding and cryptography. The design of effective multipliers, dividers, and exponentiators for finite field arithmetic is of pleasant realistic trouble. We explore and classify algorithms for finite field multiplication, squaring, and exponentiation into least significant bit first (LSB- first) scheme and most significant bit first (MSB-first) scheme, and put into effect these algorithms using semi-systolic arrays. For finite field multiplication and exponentiation, LSB-first algorithms are extra efficient as their normal cells have much less critical route computation time. An extra knowledge of LSB-first scheme is its capacity of attaining substructure sharing amongst multiple operations, which could result in financial savings in hardware when these arithmetic units are used as building blocks for a large approach [9].
Fault-centered cryptanalysis has been developed to comfortably destroy both private-key and public-key cryptosystems, making amazing finite discipline multiplication an awfully foremost research subject in latest years. As a result, this investigation presents a semi systolic Gaussian ordinary basis multiplier. Headquartered on the proposed Gaussian natural basis multiplier, both concurrent error detection and correction capabilities can be easily completed utilizing time redundancy technological know-how and not using a hardware amendment. This be taught builds a semi systolic variety-t GNB multiplier over GF (2M). No systolic array architecture for GNB multiplier has been located in the literature. The proposed semi systolic GNB multiplier is suitable for VLSI chip implementation. The proposed semi systolic GNB multiplier saves about 6 percent area complexity and 27 percentage time complexity when in comparison with present systolic foremost natural foundation multipliers of variety 2. The proposed GNB multiplier may also be modified to a multiplier with concurrent error detection and correction without a

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

hardware change of the systolic array itself. The proposed GNB multiplier with concurrent error detection most effective has zero.6 percent more space and 1 percent more time complexity than the proposed multiplier without concurrent error detection for the case of m ¼ 233 and t ¼ 2, and presents equality checkers [7].

Multiplication and squaring are most important finite subject operations in quick Fourier develop into computations and designing efficient multipliers and squares influence the efficiency of cryptosystems. We don't forget the CSE situated Montgomery multiplication in the binary extension fields and study special structures of bit-serial and bit-parallel multipliers. For each of those constructions, we learn the role of the 1st viscount montgomery of alamein aspect, and then by using utilizing correct motives, advocate new architectures. Above all, we propose two bit-serial multipliers for general irreducible polynomials, and then derive bit-parallel Sir Bernard Law multipliers for 2 most important courses of irreducible polynomials. On this regard, first we keep in mind trinomials and provide a method for locating effective 1st viscount montgomery of alamein reasons which outcome in a low time complexity. Then, we don't forget style-II irreducible pentanomials and design two bit-parallel multipliers which might be similar to the satisfactory finite subject multipliers said within the literature. Furthermore, we remember squaring making use of this family of irreducible polynomials and show that this operation can be carried out very rapid with the time complexity of two XOR gates [6].

On this paper, an efficient digit-serial systolic array is proposed for multiplication in finite field utilizing the regular foundation illustration. From the least significant bit first multiplication algorithm, we obtain a brand new dependence graph and de signal an efficient digit-serial systolic multiplier. If input information are available consistently, the proposed array can produce multiplication outcome at a price of 1 each clock cycles, the place is the selected digit measurement. Evaluation indicates that the computational extend time is significantly much less. Furthermore, due to the fact the brand new structure has the facets of regularity, modularity, and unidirectional knowledge flow, its good proper to VLSI implementation [5].

IV. PRESENT MODIFIED BOOTH MULTIPLIER ON FFT

A different development in the multiplier is by way of lowering the quantity of partial merchandise generated. The booth recording multiplier is one such multiplier; it scans the three bits at a time to lower the quantity of partial merchandise. These three bits are: the two bit from the present pair; and a third bit from the high order bit of an adjoining minimize order pair. After analyzing each triplet of bits, the triplets are changed with the aid of booth good judgment into a set of 5 manipulate signals utilized by the adder cells in the array to control the operations performed by way of the adder cells.[10,11,2]

To velocity up the multiplication sales space encoding performs a few steps of multiplication at once. Sales space's algorithm takes knowledge of the truth that an adder subtractor is practically as quick and small as a simple adder. From the fundamentals of sales space Multiplication it may be proved that the addition/subtraction operation can also be skipped if the successive bits within the multiplicand are equal. If three consecutive bits are same then addition/subtraction operation may also be skipped. Thus in many of the instances the lengthen associated with booth Multiplication are smaller than that with Array Multiplier. Nevertheless the performance of sales space Multiplier for extend is input information dependant. Within the worst case the lengthen with booth multiplier is on per with Array Multiplier. The system of sales space recording reduces the numbers of adders and hence the extend required to provide the partial sums with the aid of analyzing three bits at a time. The excessive performance of sales space multiplier comes with the hindrance of Power consumption. The intent is enormous quantity of adder cells required that consumes giant energy.

..

TABLE I COMPARISON BETWEEN MULTIPLIERS

Parameter	Array Multiplier	Wallace Tree Multiplier	Booth's Multiplier
Operation Speed	Less	High	Highest because the cycle length is as small as possible.
Time Delay	More (n+1)tFA	Log(n)	Less (ntFA/2 + ntFA)
Area	Maximum area because it uses a large number of adders.	Medium area because Wallace Tree used to reduce operands.	Minimum area because adder/subtracter is almost as small/fast as adder.
Complexity	Less complex	More complex	Most complex
Power consumption	Most	More	Less
FPGA implementation	Less efficient	Not efficient	Most efficient

Within the present architecture FFT architecture is computed centered on array multipliers because the area

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

and latency of the present approach has improved due the architectural complexity of the multipliers. The Wallace tree multiplier is noticeably turbo than a easy array multiplier on the grounds that its peak is logarithmic in word size, not linear. However, furthermore to the tremendous number of adders required, the Wallace tree's wiring is much much less typical and extra tricky. For this reason, Wallace bushes are usually refrained from with the aid of designers, at the same time design complexity is a trouble to them. So, so as the curb the demerits of the existing process we've got proposed two ways.[13,14]

V. THE PROPOSED ARCHITECTURE

A. XOR-SHIFTING Multiplier

Common *digit patterns*

$F_1 =$ 7*X = (0111)*X = X + X<<1 + X<<2

F_2 = 13*X = (1101)*X = X + X<<2 + X<<3

In order to reduce the draw backs of Bit Shifting based Multiplier: Such as area overhead and computational timing we have gone for another method which has described below.

B. Modified XOR-Shifting multiplier

Radix-4 sales space Multiplier was excellent Multiplier amongst all with less power consumption and suitable field delay alternate-off. On this sales space multiplier technique is to increase pace by means of decreasing the quantity of partial merchandise by way of half of. On account that an eight -bit sales space multiplier is used on this mission, so there are only 4 partial merchandise that need to be delivered as an alternative of eight partial merchandise generated using conventional multiplier.

									1	0	0	1	0	1	0	1	-107 (INPUT-A)
							×		0	1	1	0	1	0	0	1	105 (INPUT-B)
PARTIAL PRODUCTS	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0	1	1 × Y
	0	0	0	0	0	0	1	1	0	1	0	1	1	0			-2 × Y
	0	0	0	0	0	1	1	0	1	0	1	1					-1 × Y
	0	1	0	0	1	0	1	0	1	0							2 × Y
	1	1	0	1	0	1	0	0	0	0	0	1	1	1	0	1	-11235

Figure 2 Architecture of Booth Structure

Commonly if we multiply Eight*eight we can get eight rows of partial product but in the above structure we have now minimize 8 rows of partial merchandise into four rows i.e. rows are diminished 1/2, which helps to slash the complexity of the architecture. The proposed structure has implemented on FFT and outcome evaluation of the above method has given below

3) Polynomial based Mastrovito multiplier on Galois Field

An effective algorithm for the multiplication in GF (2m) was presented by way of Mastrovito. The space complexity of the Mastrovito multiplier for the irreducible trinomial trinomial x m + x n +1, Set of XOR, AND gates. Nevertheless XOR gates are ample when the generating trinomial is of the shape trinomial equations. We

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

additionally calculate the time complexity of the proposed astrovito multiplier, and give design examples for the irreducible trinomials x 7 + x 4 + 1 and x 6 + x three +1.Mastrovito multiplier is a category of parallel multipliers over Galois fields founded on polynomial basis representations. The polynomial groundwork of GF(pm) is given as a vector (βm-1,βm-2, …, β0) . As an illustration, remember a Galois discipline GF (28) received by way of an irreducible polynomial IP(x)=x8+x4+x3+x+1. When IP(β)=zero, a vector (β7, β6, …, β0) is a polynomial basis. Determine 1 shows the structure of Mastrovito multipliers handled in GF-ACG, which includes Matrix generator and Matrix operator. The Matrix generator first generates a matrix decided through a reminder which is given by way of the division of multiplicand via polynomial foundation.

The Matrix operator then performs the multiplication of the multiplier (i.e., a further input) and the generated matrix and sooner or later produces the product. Such Mastrovito
Multiplier is known as a GF parallel multiplier with the minimal subject rate.

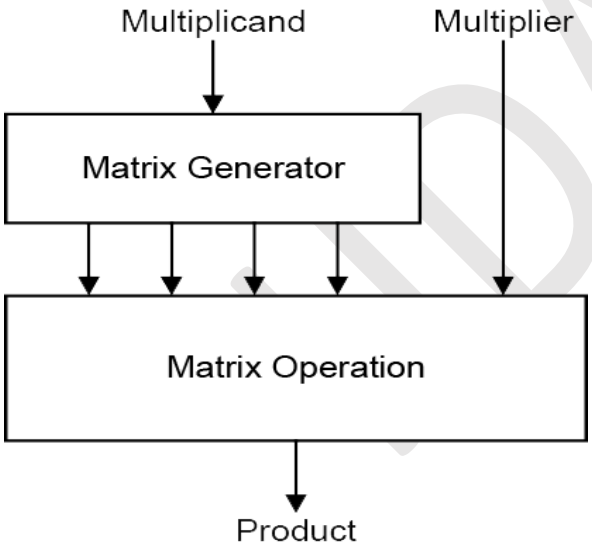

Figure 3 Architecture of a Mastrovito multiplier

VI. RESULT ANALYSIS

The highest speed will also be performed with the new architecture in designated configuration. The minimal complete time wanted for an 8bit 1st viscount Montgomery of product sinks beneath 2μs in configuration the place w is high. Regrettably this doesn't always imply that the design is prime over slightly serial design in every case. The subject consumption increases dramatically with the word size. So as make a valid assertion concerning the efficiency of a configuration, the time× subject product wants to be evaluated for every configuration. The proposed multiplier and its corresponding blocks are described making use of structural VHDL and synthesized using Xilinx Synthesis instrument (XST), net percent variant 13.2.The implementation was exact to Xilinx spartan-6 low energy, selected device: 6vlx75tlff484-1l the logical routing may also be discovered from the bought location and route result from the FPGA Editor alternative in Xilinx synthesizer. It is located that about 38% area for the certain FPGA is blanketed for the implementation of this system. The CLB's are connected in cascade manner to receive the functionality for the designed method. To ensure that the hardware implementation works appropriately, simulation experiment used to be performed utilising I-Sim
B. Influence of the Proposed go with the flow on height memory usage, Timing and subject.

In this paper, the traditional procedure and the proposed process are analyzed founded on the price function of placer and router. As proven within the (As proven within the figures which is listed in the paper) the number of LUT's, memory utilization and timing are reduced in proposed float due to less consumption of adder circuit within the design.
The routed structure of the conventional and proposed approach on Xilinx spartan-6 low Power, selected gadget: 6slx9ltqg144-1l is used and indicates the proposed method outperforms the traditional architecture indicates that the proposed process increased the combinational direction delay when in comparison with conventional approach, In terms of overhead, due to the fact the traditional approach and the proposed approach only trade the position and routing of the design, because the utilization of the CLB (configurable logic blocks) varies which supplies the overhead and delay lesser than present process. Moreover, no unreachable CLBs are stated via the fashioned process and the proposed procedure which helps to overcome the hindrance of the common approach. For this reason, the conventional technique and the proposed system preserve CLB overhead. [15, 16]

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

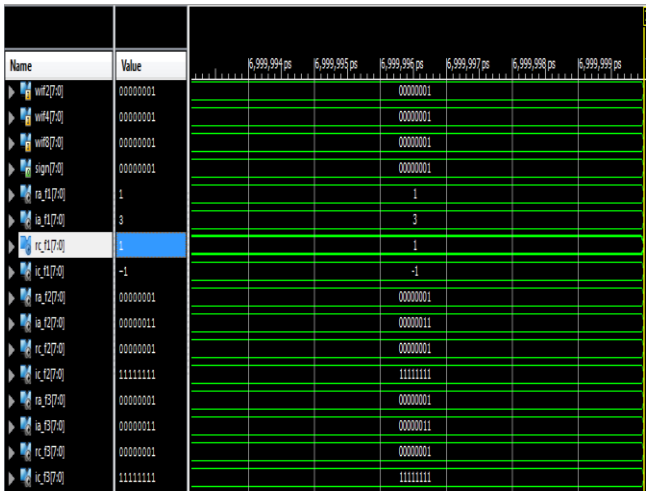

Figure 4 Simulation Results of *XOR-SHIFTING* Multiplier on FFT.

Primitive and Black Box Usage:

BELS : 2883
GND : 1
LUT1 : 12
LUT2 : 658
LUT3 : 412
LUT4 : 526
LUT5 : 38
LUT6 : 54
MUXCY : 561
VCC : 1
XORCY : 620
IO Buffers : 176
IBUF : 168
OBUF : 8

Device utilization summary:

Selected Device: 6vlx75tlff484-1l

Slice Logic Utilization:

Number of Slice LUTs: 1700 out of 46560 3%

Number used as Logic: 1700 out of 46560 3%

Figure 5 Area Analysis of *XOR-SHIFTING* Multiplier on FFT.

IO Utilization:

Number of IOs: 200

Number of bonded IOBs: 176 out of 240 73%

Total Delay 12.837ns (2.627ns logic, 10.210ns route (20.5% logic, 79.5% route)

==========

==========

Total memory usage is 273216 kilobytes

Figure 6 Timing/Memory Analysis of *XOR-SHIFTING* Multiplier on FFT.

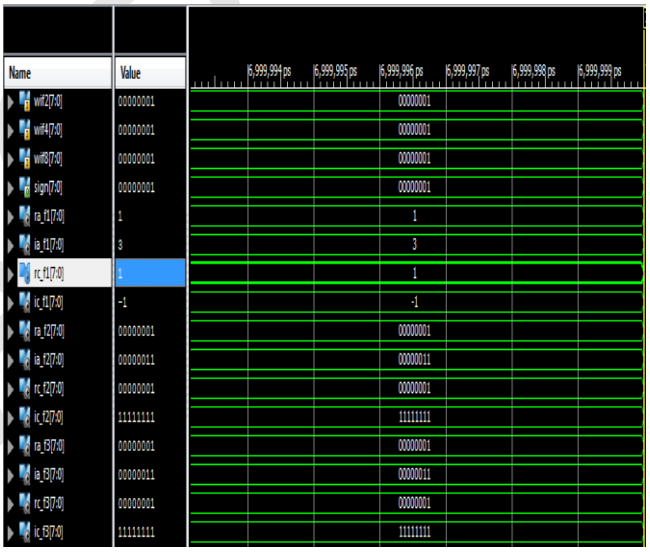

Figure 7 Simulation Results of XOR-SHIFT based Booth based Multiplier on FFT.

Primitive and Black Box Usage:

BELS : 2520
GND : 1
INV : 528
LUT2 : 193
LUT3 : 382
LUT4 : 321
LUT5 : 532
LUT6 : 292
MUXCY : 126
VCC : 1

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

```
#      XORCY                : 144
#      Flip-flops/Latches   : 1320
#      LD                   : 880
#      LDC                  : 440
#      Clock Buffers        : 16
#      BUFG                 : 16
#      IO Buffers           : 176
#      IBUF                 : 168
#      OBUF                 : 8
```

Device utilization summary:

Selected Device: 6vlx75tlff484-1l

Slice Logic Utilization:

Number of Slice Registers:	1320 out of 93120	1%
Number of Slice LUTs:	2248 out of 46560	4%
Number used as Logic:	2248 out of 46560	4%

Figure 8 Area Analysis of XOR-SHIFT based Booth based Multiplier on FFT

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs: 16 out of 32 50%

Total memory usage is 271488 kilobytes

Total delay 1.156ns (0.352ns logic, 0.804ns route) (30.4% logic, 69.6% route)

Figure 9 Timing/memory Analysis of XOR-SHIFT based Booth based Multiplier on FFT.

Cell Usage:

```
# BELS                 : 1562
#      GND              : 1
#      LUT2             : 255
#      LUT3             : 54
#      LUT4             : 902
#      MUXCY            : 154
#      MUXF5            : 19
#      VCC              : 1
#      XORCY            : 176
#      IO Buffers       : 176
#      IBUF             : 168
#      OBUF             : 8
```

Selected Device: 6vlx75tlff484-1l

Number of Slices:	680 out of 2448	27%
Number of 4 input LUTs:	1211 out of 4896	24%
Number of IOs:	200	
Number of bonded IOBs:	176 out of 158	111% (*)

Total memory usage is 254268 kilobytes

Total 31.968ns (22.115ns logic, 9.853ns route (69.2% logic, 30.8% route)

Figure 10 Timing/memory Analysis of polynomial based mastrovito GF Multiplier on FFT

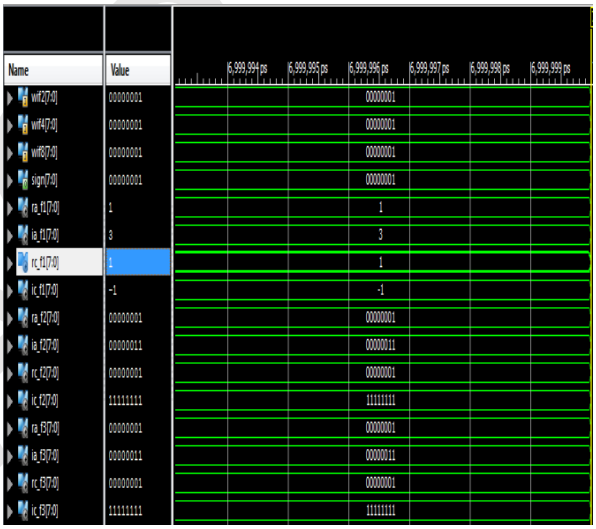

Figure 11 Simulation Results of polynomial based mastrovito on GF Multiplier on FFT.

From the above outcome it suggests the polynomial based mastrovito on GF (2m) extra effective when evaluate to different approaches. It's observed that about 38% field for the certain FPGA is included for the implementation of this procedure. The CLB's are related in cascade manner to receive the functionality for the designed process. As the insurance plan field of the CBs reduces minimize route channel width. The cut down extend comes from that the number of system defects is smaller when the elevate propagates quicker via the logic. The reminiscence usage of the proposed system is decreased up to 3.3% than the traditional approach.

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

Nonetheless, in this work the fundamental target is using the GF algorithm at the side of mastrovito in conjunction with combinational good judgment that outcome in much less subject, reminiscence and prolong. (As shown in the Figure 1 -11).

Table II Comparison between Multipliers

Methods	BELS(Basic logic elements)	Memory usage	Slices
XOR-SHIFTING	2883	273216	1700
XOR-SHIFTING on Booth	2520	271488	1320
Polynomial multiplier on mastrovito on GF(2m)	1653	254268	680

VII. CONCLUSION

In this paper, the Sir Bernard Law modular multiplication algorithm is studied and also implemented in the FPGA for obtaining a realistic assessment point of view.CSE based Montgomery multiplication algorithm is an awfully effective method of modular multiplication. The algorithm operates in binary extension fields. It includes a pre computation block, hence reducing the longest route prolong enormously. The study of the pre computation block in terms of silicon discipline concludes that the total influence is insignificant for speedy Fourier become purposes. The operand length is carried out from eight bits to 1024 bits. It can be mentioned that the based on Montgomery multiplier operates at about 1.2- 2.5 V for Spartan sequence FPGAs.

The pace and power characteristics can also be investigated and its performance in phrases of routing architecture. The outcome show that the Polynomial multiplier centered on mastrovito doubles the algorithm is four times rapid in GF (2M) mode, when compared to normal multiplication algorithms.

Conclusions of this study show that the multiplier can be increased to approach more multiplier bits in every clock cycle. The seam up power consumption of multipliers is much elegant on their switch recreation. The switch endeavor may also be diminished with the aid of graph established algorithm nevertheless it must no longer limit the velocity of the multiplier.

REFERENCES

[1] Renteria-Mejia, C.P. Lopez-Parrado, A.Velasco-Medina, J."Hardware design of FFT polynomial multipliers Circuits" and Systems (LASCAS), 2014 IEEE 5th Latin American Symposium on Year: 2014,Pages: 1 - 4,

[2] Chen, D.D. Mentens, N.Vercauteren, F.Roy, S.S.Cheung, R.C.C.Pao, D.Verbauwhede, I."High speed Polynomial Multiplication Architecture for Ring-LWE and SHE Cryptosystems"Circuits and Systems I: Regular Papers, IEEE Transactions on Year:2015, Volume: Issue: 1,Pages: 157 – 166.

[3] "Fast Fourier Transform applications of brahmaqupta-bhaskara equation," (2006) IEEE Circuits Syst. I, Reg. Papers, vol. 53, no. 7, pp. 1565–1571

[4] H.Wu, (2008) "Bit-parallel polynomial basis multiplier for new classes of finitefields,"ieeetrans.Comput.,vol.57,no.8,pp.1023–1031

[5] P. K. Meher, (2009) "On efficient implementation of accumulation in finite fieldover and its applications,"ieeetrans.verylargescale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 541–550

[6] Digital Signature Standard (DSS), FIPS 186–2, (2000) National Institute of Standards and Technology

[7] T. Zhang and K. K. Parhi, (2001) "Systematic design of original and modified mastrovito multipliers for general irreducible polynomials," IEEE Trans. Comput., vol. 50, no. 7, pp. 734–749

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

[9] P.K.Meher, (2009) "Systolic and non-systolic scalable modular designs of finitefield multipliers for reed-solomoncodec,"ieeetrans.verylarge Scale Integr. (VLSI) Syst., vol. 17, no. 6, pp. 747–757

[10] P.Montgomery, (1985) "Modular multiplication without trial division," Math, Computation, vol. 44, no. 170, pp. 519–521

[11] P. Montgomery, (2009) "Bit-serial and bit-parallel CSE based Montgomery multiplication and squaring over," IEEE Trans. Comput., vol. 58, no. 10, pp. 1332–1345

[12] K.K. Parhi, (1999) VLSI Digital Signal Processing Systems: Design and Implementation. New York: Wiley

[13] C. W. Chiou, C.-Y. Lee, A.-W. Deng and J.-M. Lin, (2006) "Concurrent error detection in CSE based Montgomery multiplication over," IEICE Trans. Fundam. Electron, Commun. Comput. Sci., vol. E89-A, no. 2, pp. 566–574

[14] S. K. Jain, L. Song, and K. K. Parhi, (1998) "Efficient semisystolic architectures for finite field arithmetic," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 6, no. 1, pp. 734–749

[15] S. B.Sarmadi and M. A. Hasan, (2000) "Concurrent error detection in finite field arithmetic operations using pipelined and systolic architectures," IEEE Trans. Comput., vol. 58, no. 11, pp. 1553–1567.