International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 15
May 2018

Design of High Speed Finite Field Multiplier Using Ppa Technique

[1] SIVAPURAPU CHOWDAIAH, [2] SHAIK MASTHAN SHARIF
[1]M.tech-Scholar, Dept of ECE, Sai Tirumala NVR Engineering College, Guntur, A.P, India
[2]Assistant Professor, Dept of ECE, Sai Tirumala NVR Engineering College, Guntur, A.P, India

ABSTRACT: In VLSI design, the performance of any system is determined by the performance of the elements i.e. Multiplier. Multiplier is one of the most important parts in any processor speed which improves the speed of the operation for example in special application processors like Digital Signal Processor (DSPs). Basically, the operational speed of any digital signal processor is strictly dependent upon the speed of the multipliers used. In this paper, parallel prefix adder and partial product generator are utilized in the multiplier for fast of operation. The proposed system provides high speed of operation. The proposed design is simulated and synthesized in Xilinx ISE. When compared with existed design the proposed design shows a significant improvement in speed.

Key words: Multiplier, finite field multiplier, partial product generator, parallel prefix adder.

I. INTRODUCTION

According to Moore's Law, for every two years the number of transistors on a chip almost doubles. For more power density and more heat on the circuits, complicated designs can be implemented on the chip. In security technologies public Key cryptography is popular and most significant one. It can provide certain unique security Services, such as key exchange and digital Signature. As mentioned above public's key Cryptography is used for the purpose of Security, they are two types (1) RSA (2) Elliptic curve. EC cryptosystem uses shorter key compared with RSA to provide the same level of Security EC used in an EC crypto system is defined over finite field's low-power Design of finite field arithmetic provides results in an EC cryptosystem. It consumes low power and more suitable for wireless application.

For hardware implementation binary Extension field denoted by GF is very attractive because it offers carry free arithmetic. There are various methods to represent field Elements in GF such as polynomial basis (PB) normal basis, and dual basis. The most popularly used basis is PB because it is adopted as one of the basis choices by organizations that set standards for cryptography applications. For efficient implementation of multipliers over GF generalized PB have been proposed. The choice of the irreducible polynomial P(x) affects the complexity of a finite field multiplier.

Irreducible polynomials have less number of non-zero terms. Irreducible polynomials can provide multipliers with lower capacity. PB finite field multiplier architectures can be categorized into bit – serial bit parallel and digit serial architecture. Bit serial architecture is area efficient, and it is too slow for many applications. Bit –parallel is fast and expensive in term of area. The digit serial architecture is flexible, it has moderate speed and reasonable cost of implementation. Two low-energy digit serial PB multipliers have been proposed binary tree structure of XOR gates are used instead of a linear array of XOR gates far degree reduction, reduce both power consumption and delay. Various digit serial multipliers were proposed Such as most significant digit, least Significant digit with modifications in architecture. A factoring technique is involved in design of a digit serial PB multiplier in GF.

II. EXISTED SYSTEM

A finite Field is defined as set of finite many elements where addition and multiplication are the operations. A binary extension field GF (2m) is generated by a degree m irreducible polynomial,

P(x) = x m +pm-1 x m -1 + ------p2 x2 +p1x+1.

P1 is either O or 1.

Dynamic power consumption in CMOS based design consists of a large number of standard cells and nets. It can be expressed as

p dynamic = p switching + p internal

Pswitching is the total switching power which Obtained by souring over all nets [a net is a connection to the cells inputs as outputs]. Switching power is the power dissipated due to the charging and discharging of the output load capacitance of a cell. P internal is the total internal power obtained by summing over all cells. The internal power of each cell is the power consumed within the cell because of the charging and discharging of internal nodes capacitances of a cell and short circuit nearest dynamic power (P dynamic) can be reduced by lowering P switching or p internal. The effective method to reduce power consumption is factoring applicable for both architecture and gate level.

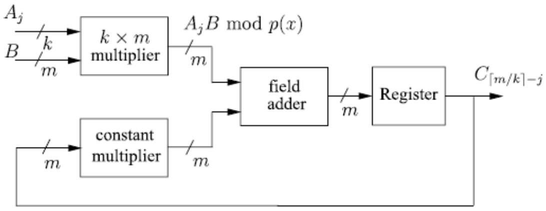

FIG. 1 FINITE FIELD MULTIPLIER

An architecture Diagram for digit serial PB multiplier in GF is shown in fig 1. There are three Modules those are k x m multiplier, and field adder. K x m Multiplier has two Operands one operand B of m-bit and others operand A j of k-bit. A j Changes for different clock cycles j. Therefore it has higher switching activity when compared with operand B.

Constant multiplier module realizes multiplication between a field element and the constant x^k field adder modules implements finite field addition using in m two –input XOR gates formed as a one layer network. Among these three k x m multiplier is the most complex module. By using this multiplier we proposed cryptography for security applications in communications.

III.PROPOSED SYSTEM

The proposed block diagram is shown in fig 1. In proposed system 'n' bits of multiplicand (Md) and multiplier (Mr) are considered. Then Mr/Md are selected. The lower bits of multiplier (Mr) are given to the selected Mr/Md. The Mr/Md performs the operation of partial product generator. Partial product generator is the combination circuit of the product generator and the 5 to 1 MUX circuit.

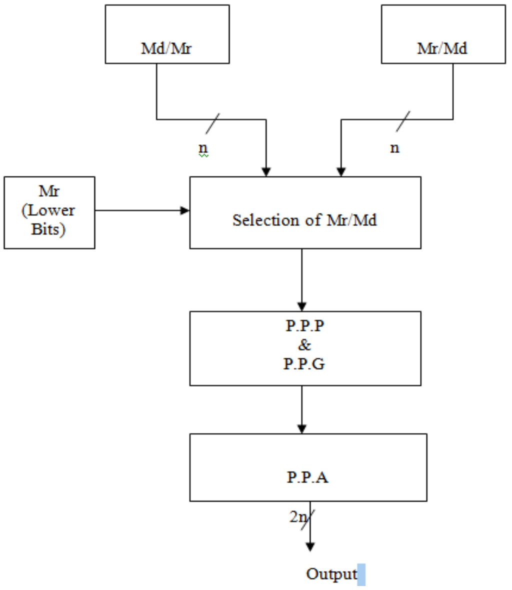

FIG 2. BLOCK DIAGRAM OF PROPOSED SYSTEM

Product generator is designed to produce the product by multiplying the multiplicand by 0, 1, -1, 2 or -2. A 5 to 1 MUX is designed to determine which product is chosen depending on the M, 2M, 3M control signal which is generated from the MBE. For product generator, multiply by zero means the multiplicand is multiplied by "0".Multiply by "1" means the product still remains the same as the multiplicand value. Multiply by "-1" means that the product is the two's complement form of the number. Multiply by "-2" is to shift left one bit the two's complement of the multiplicand value and multiply by "2" means just shift left the multiplicand by one place. The output from the partial production generator is given to the efficient parallel prefix adder.

The parallel prefix adder which is used in the multiplier is flexible to speed up the binary addition and the structure looks like tree structure for the high performance of arithmetic operations. In ripple carry adders each bit have to wait for the last bit operation. In parallel prefix adders instead of waiting for the carry propagation of the first addition, the idea here is to overlap the carry propagation of the first addition with the computation in the second addition, and so forth, since repetitive additions will be performed by a multioperand adder.

Research on binary operation elements and motivation gives development of devices. Field programmable gate arrays [FPGA's] are most popular in recent years because they improve the speed of microprocessor based on applications like mobile DSP and telecommunication. The construction of efficient parallel prefix adder consists of three stages. They are pre-processing stage, carry generation stage, post-processing stage.

A. Pre-Processing Stage

In the pre-processing stage, generate and propagate are from each pair of inputs. The propagate perform "XOR" operation of input bits and generate operation "AND" operation of input bits. The propagate (Pi) and generate (Gi) are shown in below equations 1 and 2.

$$P_i = A_i \text{ XOR } B_i ---- \quad (1)$$

$$G_i = A_i \text{ AND } B_i ---- \quad (2)$$

B. Carry Generation Stage

In this stage, carry is generated for each bit called as carry generate (Cg). The carry propagate and carry generate is generated for the further operation but final cell present in the each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. The carry generate and carry propagate are given in below equations 3 and 4.

$$C_p = P_1 \text{ AND } P_0 ------- \quad (3)$$

$$C_g = G_1 \text{ OR } (P_1 \text{ AND } G_0) -- \quad (4)$$

The above carry propagate Cp and carry generation Cg in equations 3 & 4 is black cell and the below shown carry generation in equation 5 is gray cell. The carry propagate is generated for the further operation but final cell present in the each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. This carry is used for the next bit sum operation, the carry generate is given in below equations 5.

$$C_g = G_1 \text{OR} (P_1 \text{ AND } G_0) -- \quad (5)$$

C. Post-processing stage

It is the final stage of an efficient parallel prefix adder, the carry of a first bit is XORed with the next bit of propagates then the output is given as sum and it is shown in equation 6.

$$S_i = P_i \text{ AND } C_{i-1} ----- \quad (6)$$

It is used for two sixteen bit addition operations and each bit carry is undergoes post-processing stage with propagate, gives the final sum. The first input bits goes under pre-processing stage and it will produce propagate and generate. These propagates and generates undergoes carry generation stage produces carry generates and carry propagates, these undergoes post-processing stage and gives final sum.

.

IV.RESULTS

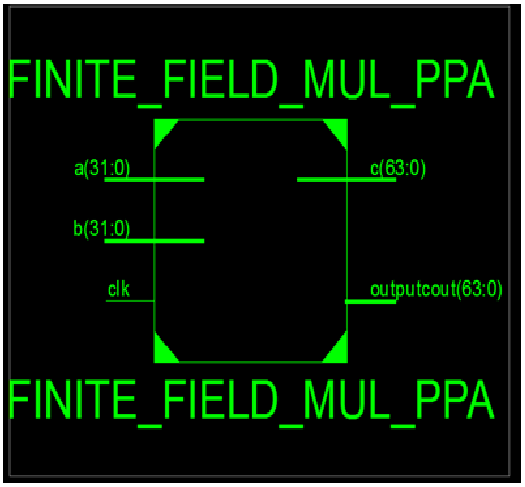

FIG 3. RTL SCHEMATIC

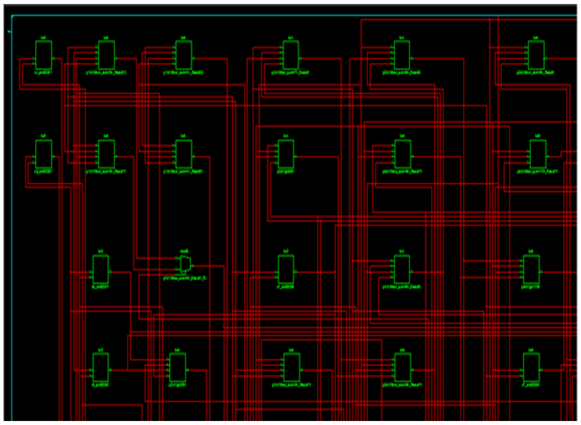

FIG 4. TECHNOLOGY SCHEMATIC

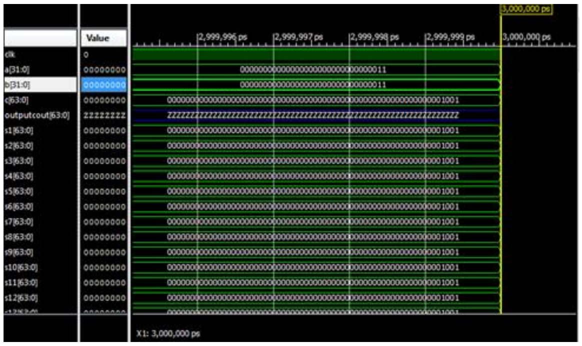

FIG 5. OUTPUT WAVEFORM

V.CONCLUSION

In this paper, we propose Multiplication operation is performed which is better performance than existed multiplier. We utilize the parallel prefix adder and partial product generator for fast of operation. The proposed multiplier is fast and efficient because of the parallel prefix adder design which does the carry propagation quickly. The required hardware and the chip memory reduces and it reduces delay i.e., speed is increased. The multiplier architecture and fast performance makes this particularly attractive for VLSI implementations.

VI.REFERENCES

[1] C. F. Kerry, "Digital signature standard (DSS)," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, FIPS PUB 186-4, 2013.

[2] *IEEE Standard Specifications for Public-Key Cryptography*, IEEE Standard 1363-2000, Aug. 2000, pp. 1–228.

[3] H. Fan and Y. Dai, "Fast bit-parallel *GF*(2*n*) multiplier for all trinomials," *IEEE Trans. Comput.*, vol. 54, no. 4, pp. 485–490, Apr. 2005.

[4] A. Cilardo, "Fast parallel *GF*(2*m*) polynomial multiplication for all degrees," *IEEE Trans. Comput.*, vol. 62, no. 5, pp. 929–943, May 2013.

[5] T. Beth and D. Gollman, "Algorithm engineering for public key algorithms," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 4, pp. 458–466, May 1989.

[6] L. Song and K. K. Parhi, "Efficient finite field serial/parallel multiplication," in *Proc. Int. Conf. Appl. Specific Syst., Archit. Processors (ASAP)*, Aug. 1996, pp. 72–82.

[7] M. Nikooghadam and A. Zakerolhosseini, "Utilization of pipeline technique in AOP based multipliers with parallel inputs," *J. Signal Process. Syst.*, vol. 72, no. 1, pp. 57–62, Jul. 2013.