AREA AND DELAY EFFICIENT DESIGN FOR PARALLEL PREFIX FINITE FIELD MULTIPLIER

[1] CH.JAYA PRAKASH,[2] P.HAREESH, [3] SK. FARISHMA
[1&2]Assistant Professor, Dept. of ECE, [3]M.Tech-Student,
Sir CR Reddy College of Engineering, West Godavari (Dt), Eluru, AP.

ABSTRACT: Multiplication is the basic building block for several DSP processors, Image processing and many other. Over the years the computational complexities of algorithms used in Digital Signal Processors (DSPs) have gradually increased. This requires a parallel prefix multiplier to achieve high execution speed or to meet the performance demands. The architecture of proposed multiplier mainly consists of parallel prefix Adder. Adders are most significant in digital signal processing and control systems. The high-speed and a processor or system accuracy is based on the performance of adder. The parallel prefix adder consists of three stages of operations they are pre-processing stage, carry generation stage, post-processing stage. This adder makes the proposed multiplier fast of operation. In this research work, a new design of Parallel Prefix Multiplier is proposed and this proposed design of multiplier uses a very fast parallel prefix adder. The experimental results show that our proposed system reduces the area by 29484Kbytes and delay by 64.825ns.

Key words: Parallel prefix adder, proposed multiplier, gray cell, finite field multiplier

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.

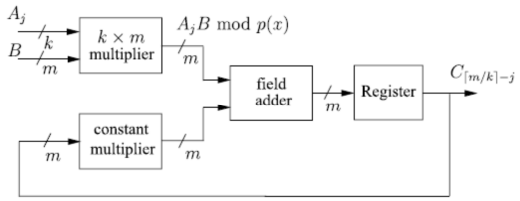

Fig. 1 Finite Field Multiplier

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

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.

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

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

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.

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

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

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.

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

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.

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

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. The step by step process of parallel prefix adder is shown in Fig 2.

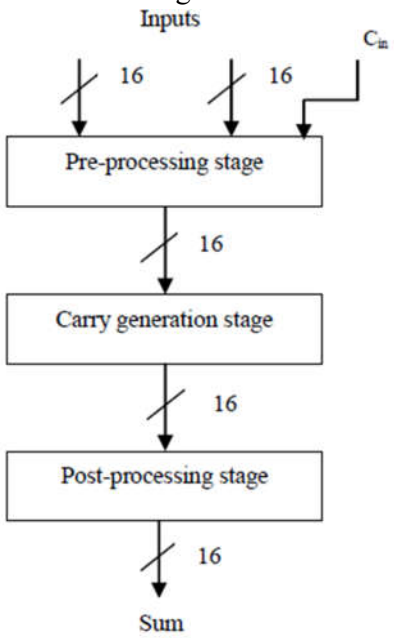

Fig 2: Flow chart for PPA

In Efficient PPA, black cell operates three gates and gray cell operates two gates. The gray cell reduces the delay and memory because it operates only two gates. The

proposed adder is design with the both black and gray cells. By using gray cell operations at the last stage of proposed adder gives a enormous dropping delay and memory used.

In Parallel Prefix adders the execution of an operation is in parallel. This is done by segmentation the operation in smaller pieces which are computed in parallel. The output is depends on the initial inputs. Parallel Prefix Adder (PPA) is equivalent to carry look ahead adder (CLA). A Carry look ahead adder is a type of adder used in digital logic. CLA is designed to overcome the latency introduced by repelling effect of carry bits in RCA. A CLA improves speed by reducing carry bits. It calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of larger bit value.

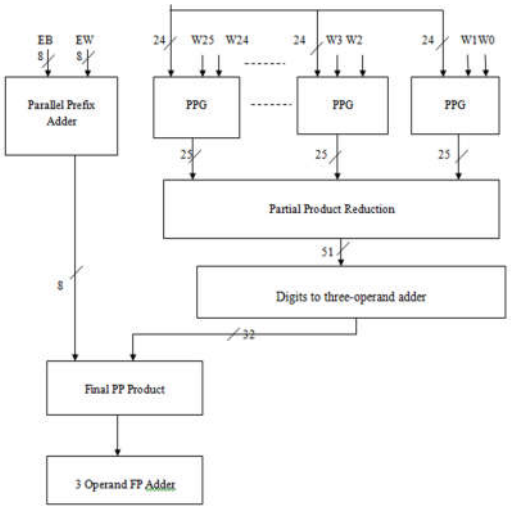

Fig. 3: Proposed Redundant PP Multiplier

CLA uses the concept of generating (G) and propagating (P) carries. These two are differ in the way their carry generation block is implemented. The main advantage of PP Multiplier is the carry reduces the number of logic levels by essentially generating the carries in parallel. PP finite field multiplier is fastest multiplier with focus on design time and is the choice for high performance multiplier in industry.

IV. RESULTS

TABLE 1
Comparison Table of area and delay

	EXISTED SYSTEM	PROPOSED SYSTEM
AREA	371744	342260
DELAY	110.363	45.538

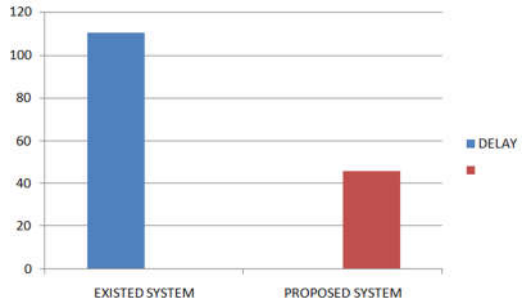

Fig 4. Comparison graph of delay

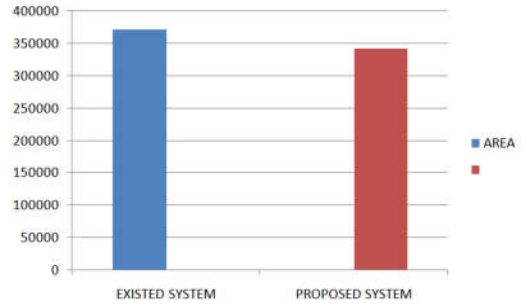

Fig 5. Comparison graph of area

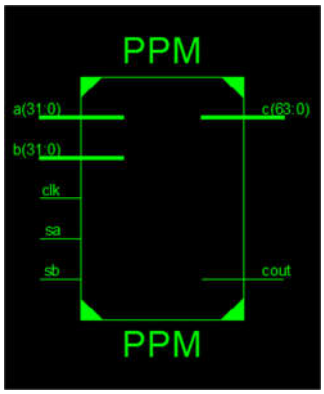

Fig 6. RTL Schematic

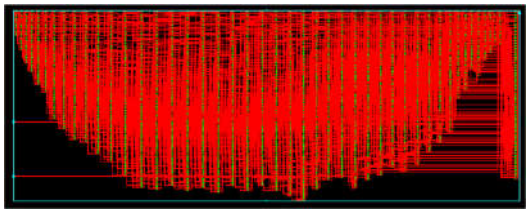

Fig 7. Technology Schematic

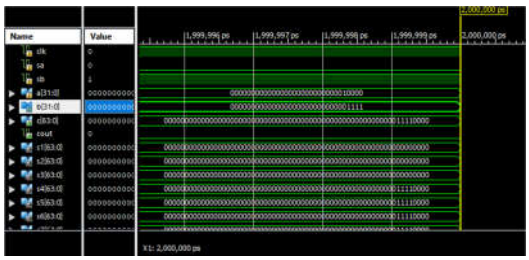

Fig 8. Output Waveform

V. CONCLUSION

In this project, An Multiplier using Parallel Prefix Adder is proposed. The latency of existing multiplier has been reduced. The conditional sum technique used in the adder is a good technique for energy efficiency. The reduction in energy of the carry tree is more than the energy overhead due to conditional sum block due to the fact that the sum block complexity is less than that of carry tree. It is proved that the adder structure implemented in this work has reduced delay. However, the parallel prefix finite field multiplier has better energy efficiency. The results prove that the proposed architecture is more efficient than the conventional one in terms of memory used and speed. The results show that in existed system, area occupied is 371744Kbytes and delay is 110.363ns. In our proposed system, area occupied is 342260Kbytes and delay is 45.538ns. Hence, by the results our proposed system is more efficient than existed system.

V. REFERENCES

[1] S.-L. Lu, “Speeding up processing with approximation circuits,” Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.

[2] J. Han and M. Orshansky, “Approximate computing: an emerging paradigm for energy-efficient design,” in Proc. ETS, pp. 1-6, May2013

[3] C. Labrado, H. Thapliyal and F. Lombardi “Design of Majority Logic Based ApproximateArithmetic Circuits,” inProc. IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2122-2125, May 2017.

[4] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, “Impact: Imprecise adders for low-power approximate computing,”in Proc. Int.Symp. Low Power Electronics and Design (ISLPED) , pp. 409– 414, Aug. 2011.

[5] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, “Approximate xor/xnor-based adders for inexact computing,” in Proc. 13th IEEE Conf. Nanotechnology (IEEE-NANO), pp. 690–693, Aug.2013.

[6] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. “Low-power digital signal processing using approximate adders,”IEEE Trans. Comput.-Aided Des. Integ. Circuits Syst, vol. 32,pp. 124–137, 2013.

[7] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, and J. Henkel.

"Architectural-Space Exploration of Approximate Multipliers,". in Proc. Int. Conf. Comput.-Aided Des. (ICCD), pp. 1-6,Nov.2016.

CH JAYA PRAKASH completed his B.Tech in ECE from JNTU Hyderabad and M.Tech in VLSI System Design from JNTU Kakinada. He is working as Assistant Professor in the Dept. of ECE, Sir CR Reddy College of Engineering, Eluru.

HAREESH PANCHETI completed his B.Tech in ECE from JNTU Hyderabad and M.Tech in VLSI Design from SASTRA University. He is working as Assistant Professor in the Dept. of ECE, Sir CR Reddy College of Engineering, Eluru.

SK. FARISHMA completed her B.Tech in BVSR engineering college, Chemakurthy and pursuing M.Tech in Sir CR Reddy college of engineering, Eluru. Her area of interest is VLSI.