Volume: 5 Issue: IV Month of publication: April 2017

DOI: http://doi.org/10.22214/ijraset.2017.4058

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

FPGA High Performance Pipelined Architecture Of Elliptic Scalar Multiplication Over GF(2m) for IOT

Kiran Sulthana S[1], Dr. V. Ellappan[2]

[1]PG scolar, [2]Professor, Department of ECE, Mahendra Engineering College, Namakkal, India.

Abstract*: The existing era has seen an excellent growth in communications. Various applications like Internet banking, mobile communication, Personal Digital Assistasnce(PDAs), smartcards, etc. have emphasized the need for security in resource constrained environments. Elliptic curve cryptography (ECC) serves as a perfect cryptographic tool as it has short key sizes and security comparable to that of other standard public key algorithms. The efficiency is largely affected by the underlying arithmetic primitives. Multiplication and Inversion are the two most important primitive fields which FPGA design of our concept reveals. The smallest programmable entity in an FPGA is the look up table. The arithmetic algorithms proposed in this thesis maximizes the utilization of LUTs on the FPGA. A novel finite field multiplier based on the recursive Karatsuba algorithm is proposed. The proposed multiplier combines two variants of Karatsuba, The general Karatsuba multiplier has a large gate count but for small sized multiplications is compact because it utilizes LUT resources efficiently. For large sized multiplications, the simple Karatsuba is efficient as it requires lesser gates. In our hybrid multiplier, the simple algorithm performs the initial recursion whereas the general algorithm performs the final multiplication of small sizes. The multiplier thus obtained has the best area time product compared to reported literature.

***Keywords*: Cryptography, Elliptic Curve Cryptography, FPGA, Karatsuba Multiplier, Recursion**

I. INTRODUCTION

Elliptic curve cryptography is rapidly become the standard for public key ciphers because of the large amount of security provided per key bit. Many authorised concerns have shifted from other public-key cryptographic techniques to ECC. To match the speed requirements for real time applications hardware acceleration of ECC is a necessity. For the inversion another important field primitive, the ItohTsujii algorithm (ITA), is designed to use optimal exponentiation circuits specific to the LUT size of the underlying FPGA platform. These field primitives are combined to realize the elliptic curve scalar multiplier. This project explores opportunities for pipelining the design for high-speed. This project extends the work proposed in to approximate the delay in the critical path by the number of k input LUTs in the path. This is used to determine analytically the location of the pipeline stages in the architecture. The number of pipeline stages in the architecture also critically affects the computation time. The optimal number of pipeline stages in the design. The framework has been applied on two well-known scalar multiplication techniques, namely, left-to-right double and add algorithm with binary signed digit representation of the scalar, and the Montgomery ladder based scalar multiplication. The analysis shows that, while a three stage pipeline gives best performance for the former scalar multiplication method, a four stage pipeline is more suited for the latter. The three- stage pipelined architecture for double and add based scalar multiplication on Xilinx Spartan 3E platforms has a computation time of 10 µs with an area of 3789 slices. The four-stage pipelined architecture needs a computation time of 9.5 µs, with an area of 3513 slices on the Spartan 3E platform, when this technique is applied on the scalar multiplier based montogomery ladder.

II. RELATED WORK

Literature is a significant treasure house of various VLSI architectures for point multiplication in ECC. The followings are the various literatures regarding the exsisting architectures. Reza Azarderakhsh and Koray Karabina [13] designed a new double point multiplication algorithm and its application to binary elliptic curves with endomorphism. The differential addition chains based algorithm was written in this design. The architecture was designed with a uniform structure and has some degree of built-in resistance against side channel analysis attacks. Its double point multiplication algorithm depends on an adaptation of Montgomery‘s PRAC algorithm.

Efficient elliptic curve point multiplication using digit-serial binary field operations was designed by Gustavo D. Sutter et.al [14]. They used a new high-speed point multiplier for elliptic curve cryptography using either field programmable gate array or application-specified integrated circuit technology. Their design adapted a digit-serial approach in GF multiplication and GF division in order to construct an efficient elliptic curve multiplier using projective coordinates. The design involved many basic arithmetic operations in the underlying finite field. There are different acceleration techniques to improve the performance of the ECC operations. Three types of algorithm Montgomery Ladder Algorithm are used in the point multiplication technique. Point multiplication using three multipliers and one divisor and precomputing xP−1. This design achieved point multiplication over GF (2163) in 19.38 µs in Virtex-E devices and in 5.48 µs in Virtex-5.

Efficient RNS implementation of elliptic curve point multiplication over GF (p) was designed by Mohammad Esmaeildoust et.al [15]. In this design, based on the residue number system (RNS), new hardware architecture for ECPM over GF (p) was established. The designed architecture encompasses RNS bases with various word-lengths to efficiently implement RNS Montgomery multiplication. In that method two versions of fast and area-efficient designs for RNS Montgomery multiplication in six and four-stage pipelined architectures were used. When compared to state-of-the-art implementations, their implemented design achieved higher speeds and better area–delay.

III. METHODOLOGY

A. FPGA – Based Elliptical Curve Cryptographic Coprocessor

Elliptic curve cryptosystems are public key protocols whose security is based on the conjectured difficulty of solving the discrete logarithm problem on an elliptic curve. Assuming Q to be a point of order n on an elliptic curve it is desirable to compute mQ, where m is an integer smaller than n. This will be done by using several additions, doublings, or possibly negations of points on the elliptic curve to achieve the result. These operations depends to arithmetic operations in the finite field K = Fqn, over which the elliptic curve definition happens. The fields which have characteristic 2, i.e., q is a power of 2 has been concentrated more in this work. The required computations to compute mQ can be categorized at three levels. Each requires thorough investigations to enable the design of a high performance elliptic curve co-processor.

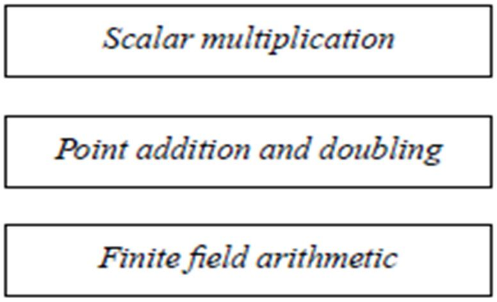

Fig 1 Three Stages of Performing Elliptic Cuve

B. Processing Unit

The processing unit is a combined architecture for differential point addition and differential point doubling operations. The major portion of the available slices are occupied by the processing unit because of the involvement of various finite field arithmetic units for computing the output point addition and doubling values. So the main contribution of this work is focused on designing an area efficient processing unit with a reduction in number of incorporated Arithmetic units. The modified area efficient data dependency graph for the processing unit. Proposed data dependency graph for computing double point multiplication employs area efficient finite multipliers, squarers, and adders based on differential point addition and doubling formulae given in [4]. For reducing the arithmetic units for computation we have designed the processing unit using 4 stage of pipeline process. The inputs to the processing unit are three points and a difference between two points (the input points values are selected based on the sequence from the control unit). The parameter _a‘ is a constant integer value from the elliptic curve equation considered for cryptography.

Four stages of input point processing happens when the processing unit is loaded with input. After the completion of the previous stage, the values are stored temporarily in the respective registers (Buffers) and then only the next level of process begins. Therefore in the proposed architecture, the reuse of registers and other arithmetic units that are used in previous stage of process takes place.

For example in data flow graph the multiplier used in the first stage of computation can be reused in the stage four and the squarer used in the fourth stage can be reused in the final output computation stage thereby reducing the need for extra multipliers and squarers. The buffers that are used in the previous stage and that are found to be empty in the next stages are reused efficiently for making processing unit area efficient The arithmetic units that are incorporated inside proposed resource reusable combined architecture for differential point addition and differential point doubling are discussed in detail in following sections.

C. Multiplication Unit

The design of Finite field multipliers is the complex issue in the implementation of the ECC processor. A number of multipliers with different area and time complexity are reported in the available literatures. In this work, an area efficient architecture for Karatsuba‘s multiplier which incorporates digit-level polynomial basis multiplier is adopted. The modified Karatsuba multiplier used in proposed architecture for double point multiplication multiplies 2 finite inputs _A‘ and _B‘ of m-bit length. Processing in Karatsuba multiplier is preceed with the splitting of each operand into two equal parts.The internal processor includes 3 multipliers and 4 adders. The architecture for Karatsuba multiplier is as shown in figure 2.

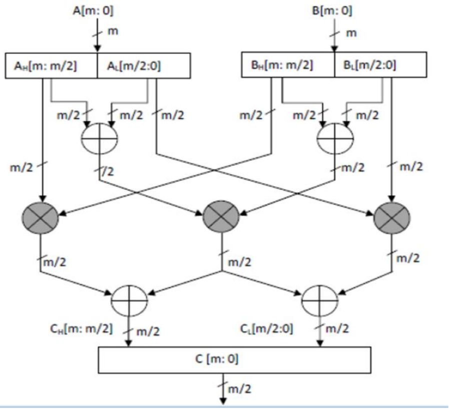

Fig 2Architecture for *G(2m)* Karatsuba multiplier .

IV. RESULT AND COMPARISION

We synthesized our design on a Xilinx Virtex-6 XC6VSX475T-2 FPGA using ISE 11.5. Karatsuba multipliers are constructed by applying Karatsuba layers recursively down to 32-bit multipliers generate by Xilinx coregen. We found that a 32-bit fast carry-chain has a maximum clock frequency of 400 *MHz*, which is close to the maximum frequency of the 32-bit multipliers. Thus, we select 32-bits as our limb-width in every Karatsuba layer. Figure 4 shows the resource utilization of batch pipelined Montgomery multipliers with different bitwidths.

We found the designs’ clock frequencies are lower than the theoretical 400 *MHz* limit. A major contributor to this is likely increased routing delay in large designs as the average separation between components increases. The insertion of additional pipelining registers reduce this routing delay. We implemented software Montgomery multiplication using Algorithm 1 with the long integer multiplication functions provided by the GMP multiple-precision library [14]. We benchmarked this software implementation on an Intel Xeon E5420 CPU running at 2.5 *GHz*. The single-threaded performance was multiplied by 4 to estimate the maximum aggregate performance of using all four cores. The estimation of power consumption is done by the CPUs thermal design power (TDP) specification.

I estimate the maximum number of different bit-width multiplier cores that could be mapped to the Virtex-6 and their performance. The power consumption of these FPGA implementations is estimated using the Xilinx power estimator tool with an activity rate of 50 %, a very pessimistic estimate. Table I shows the performance speedup and per-multiplication energy efficiency of our FPGA implementations against the software implementations. We normalized the performance and resource requirements of previous 512-bit block-wise Montgomery multiplier designs with the architecture proposed in this paper. It is impossible to determine the exact operating frequency of these designs, we optimistically assume they can all operate at 500 MHz when mapped to our target Virtex-6 device. We also normalized the fine-grained logic usage of [10] to the Virtex-6 LUT by assuming that each Virtex-6 LUT can realize one Virtex-2 slice. We calculated both the normalized area*L*-delay product (i.e.

LUTs) and area*M*-delay product (i.e. embedded multiplier) as shown in Table II. We achieve significantly lower areadelay products thanks to the $O(N(\log 3= \log 2))$ complexity of the Karatsuba algorithm.The size of our Karatsuba-based Montgomery multiplier is major drawback. Large multipliers found it difficult to operate at high clock frequencies because of routing delay eventhough high throughput is provided by fully parallelized . This can be overcomed in future works by finding different ways of serializing parts of the architecture.

V. CONCLUSIONS

In this paper, we presented a Karatsuba-based Montgomery multiplier for cryptography applications using long integers. The multipliers have significantly lower area-delay products compared with previous designs. They also provide excellent performance and energy efficiency compared with software implementations. Future work includes research on serializing the design so that it can reach parts of the design space using fewer resources.

The simple Karatsuba multiplier is more efficient for large sized multiplications. After a thorough search, a threshold of 29 was found. Multiplications smaller than 29 bits is done using the classical Karatsuba multiplier, while larger multiplications are done with the the bit parallel Karatsuba multiplier.

REFERENCES

[1] K. Järvinen, “Optimized FPGA-based elliptic curve cryptography processor for high speed applications,” Integr., VLSI J., vol. 44, no. 4, pp. 270–279, Sep. 2011.

[2] J. A. Solinas, “Efficient arithmetic on Koblitz curves,” Designs, Codes Cryptography, vol. 19, nos. 2–3, pp. 195–249, 2000.

[3] K. Järvinen and J. Skytta, “On parallelization of high-speed processors for elliptic curve cryptography,” IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 16, no. 9, pp. 1162–1175, Sep. 2008.

[4] R. Azarderakhsh and A. Reyhani-Masoleh, “Efficient FPGA implementations of point multiplication on binary edwards and generalized Hessian curves using Gaussian normal basis,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. PP, no. 99, p. 1, Jun. 2011.

[5] N. A. Saqib, F. Rodríguez-Henríquez, and A. Diaz-Perez, “A parallel architecture for fast computation of elliptic curve scalar multiplication over GF(2m),” in Proc. 18th Int. Parallel Distrib. Process. Symp., Apr.2004, pp. 144–151.

[6] Q. Pu and J. Huang, “A microcoded elliptic curve processor for GF(2m)using FPGA technology,” in Proc. Int. Conf. Commun., Circuits Syst.,vol. 4. Jun. 2006, pp. 2771–2775.

[7] Xilinx, “Using block RAM in spartan-3 generation FPGAs,” Xilinx, San Jose, CA, Tech. Rep. XAPP-463, 2005.

[8] K. Sakiyama, L. Batina, B. Preneel, and I. Verbauwhede, “Superscalar oprocessor for high-speed curve-based cryptography,” in Proc. 8th Int.Workshop Cryptographic Hardw. Embedded Syst., 2006, pp. 415–429.

[9] B. Ansari and M. Hasan, “High-performance architecture of elliptic curve scalar multiplication,” IEEE Trans. Comput., vol. 57, no. 11, pp.1443–1453, Nov. 2008.

[10] W. N. Chelton and M. Benaissa, “Fast elliptic curve cryptography on FPGA,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 2,pp. 198–205, Feb. 2008.