ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 5, March 2017

National Conference on Advanced Computing, Communication and Electrical Systems - (NCACCES'17)

24th - 25th March 2017

Organized by

C. H. Mohammed Koya
KMEA Engineering College, Kerala- 683561, India

A Comparison of High-Speed Polynomial Multiplication Algorithms for Modern Cryptosystem

Naizam P A[1], Prof. Riyas K S[2]

M. Tech Student, Dept. of Electronics and Communication Engineering, RIT Kottayam, Kerala, India[1]

Associate Professor, Dept. of Electronics and Communication Engineering, RIT Kottayam, Kerala, India[2]

ABSTRACT: In the present world, applications of digital communication have increased day by day. Hence more care should be taken with security issue of digital communication network such as online money transaction, data communication over internet, LAN, MAN network etc. So, it is very helpful to transmit information via secure network. So, it is essential to make a crypto-system which enable encryption and decryption of data. This Paper explores efficient hardware architectures for polynomial multiplication for modern cryptography like Ring-LWE, SHE etc. The hardware Architectures are based on FFT algorithm. All architecture uses polynomial modular multiplication based on FFT which makes the processing time faster and used comparatively smaller amount of space in the FPGA. Each block is coded with Verilog high description language and it is synthesized and simulated using Xilinx-ISE14.7.

KEYWORDS: Fast Fourier transform, FFT based multiplication of large numbers, Wingord algorithm, Karatsuba algorithm, Modular reduction algorithm, FPGA

I.INTRODUCTION

In the present day, cryptography is the most essential part of a system, where information security has become more and more important area of interest. Cryptography prior to the modern age is defined as a combination of conversion of information from readable state to apparent unknown state (called Encryption) and conversion of information from unknown state to known or readable state (called Decryption) [1]. Modern cryptography is heavily based on complex mathematical algorithms and it is theoretically possible to break the information, but practically quite impossible one due to complexity of algorithms. The growth of cryptographic technology introduces variety of security system which is essential to transfer large amount of data in secure mode with advantages of improvements in integer factorization algorithms, faster computing technologies etc. [2].

Cryptography involve creation of cipher key and combine it with original data which create an encrypted data and sent it to receiver and receiver will decrypt received data and retrieve original data. Classical cryptography like RSA and elliptic curve cryptography (ECC) [6] are facing the problems of hardness of factoring, Elliptic Curve Discrete Logarithm Problem etc. [2]. However, it can be solved by a powerful quantum computer. But it is unclear whether a sufficient powerful quantum computer can be built with the fast improvement of cryptanalysis, computation power, and the unpredictable development of the quantum computer, post-quantum security etc. So, an alternative is needed [2].

Lattice-based cryptosystems are alternative method for replacing these classical cryptosystems, because of their post quantum security proofs and the fact that, there is no known quantum algorithm that can solve the lattice problem efficiently. Variety of encryption are developed by researcher. Some examples for lattice based cryptography are Ring-LWE (Learning with error) and SHE (Somewhat Homomorphic Encryption). Beside the significant progress in the theory of lattice based cryptography, practical implementation issues are gaining attention from the research community

[2]. One important achievement in lattice based cryptography is that it reduces the gap between theory and practical by introducing ideal lattice theory. It reduces the size of parameter set as well as computational time. However, due to the large key size and the complicated computation, there is still possibility of research [2].

Since multiplication operation is very crucial and time consuming operation in any cryptographic system. We should care about multiplication which increases both speed and efficiency of entire System. Since modern world, data should be transmitted within few milliseconds arises a requirement of high speed multiplier architecture [4].
In order to facilitate researcher's hardware designs, a versatile and efficient polynomial architecture will give great help. Initial hardware design for such an encryption was fully parallelism based approach. One of the drawback of such a system is larger and complex circuitry. So, it consumes large area and the throughput of the encryption outperform the software implementation by a factor of 316.So researcher found an alternate solution that is based on FFT/IFFT approach. Such approach is targeted area-efficiency design approach [3].

The rest of this paper is organized as follows. Section II describe mathematical preliminaries and explain some theoretical concept about NTT and FFT [5], wingord [7], Karatsuba and Modular Reduction Algorithms [11]. Section III present the comparison result and discussion part. And finally, Section IV concludes this paper.

II. MATHEMATICAL PRELIMINARIES

Consider the ideal lattice $Zp/f(x)$ for some irreducible polynomial f(x) of degree n where f(x) is defined as $f(x) = f_0 + f_1x + f_2x^2 + ... + f_{n-1}x^{n-1}$
Suppose a(x) 1and b(x) are two polynomials, the multiplication of a(x) and b(x) by using the school-book algorithm is calculated as

$$a(x).b(x) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i b_j x^{i+j} \ mod \ f(x) \tag{1}$$

with a quadratic complexity, $O(n^2)$ [4]
In this paper, consider on commonly used case $f(x)=x^n + 1$ n is a power of 2, and p is a prime number with $p = 1 \ mod \ 2n$. The choice of $f(x)=x^n + 1$enables the usage of the property $x^n = -1$ which can
Simplify the polynomial multiplication as

$$a(x).b(x) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} (-1)^{\frac{i+j}{n}} a_i b_j x^{i+j \ mod \ n} \tag{2}$$

Using this property, the multiplication still has quadratic complexity with O(n2) multiplications and (n-1)2 additions or subtractions. [3]

A. NUMBER THEORETIC TRANSFORM AND FAST FOURIER TRANSFORM

The Number Theoretic Transform (NTT) is a discrete Fourier transform defined over a finite ring $Zp = Z/pZ$ Let ω be a primitive n-th root of unity in Zp, and a(x), A(X) be polynomials of degree less than n. The n-point NTT are defined as:

$$A_i = NTT_w^n(a(x))^i = \sum_{j=0}^{n-1} a_j \omega^{ij} \ mod \ p \tag{3}$$

where i=0,1, 2,....,n-1. The FFT is a DFT algorithm which reduces the number of computations needed for N points from $O(N^2)$ to O ($N \log N$) where log is the base-2 logarithm.
Using FFT we can compute convolution efficiently, thereby we can compute multiplication of two polynomials by zero padding. In order to enable two FFT computations in parallel, the smallest architecture consists of two butterflies together with two multipliers [8]. These processing units can compute two coefficients for FFT($\bar{a}$) and FFT($\bar{b}$) concurrently. IFFT is different from FFT in the multiplication by ω^{-i} instead of ω^i, and an additional multiplication by n^{-1} at the end of the last stage. Hence, the processing units of the FFT can be fully reused to compute the IFFT. Since only one IFFT instead of two is required, the two butterflies and multipliers are all used to compute IFFT($\bar{c}$) in parallel, which doubles the speed for the IFFT [4].
Algorithm for polynomial multiplication using FFT:
Input: a, b
Output:c
Step 1. A=FFT(a,ωm)
Step 2. B=FFT(b,ωm)
Step 3 for i=0 to 2n-1 do
Step 4 C[i]=A[i]. B[i]
Step 5 end
Step 6 c=FFT−1(C,ωm)
Step 7 return c

B. WINOGRADS FOURIER TRANSFORM ALGORITHMS(WFTA)

The Winograd Fourier Transform Algorithm (WFTA) is a technique that combines the Raders index mapping and Winograds short convolution algorithm for prime-factors into a composite-N Fourier Transform structure with fewer multipliers. Rader's algorithm (1968) is a fast Fourier transform (FFT) algorithm that computes the discrete Fourier transform (DFT) of prime sizes by re-expressing the DFT as a cyclic convolution [3]. Since Rader's algorithm only depends upon the periodicity of the DFT kernel, it is directly applicable to any other transform (of prime order) with a similar property, such as a number-theoretic transform or the discrete Hartley transform [9].

Winograd's small convolution algorithm performs the convolution with minimum number of multiplications and additions and thus the computational complexity of the process is greatly reduced [7]. Cyclic convolution is also known as circular convolution. Let h={h_0,h_1,h_2,....,h_{n-1}} be the filter coefficients and x={x_0,x_1,x_2,....,x_{n-1}} be the data sequence. The cyclic convolution can be expressed as

$$s(p) = h(p)x(p) \bmod (p^n - 1) \quad (4)$$

The Winograd algorithm, factorizes $z^N - 1$ into cyclotomic polynomials these often have coefficients of 1, 0, or 1, and therefore require few (if any) multiplications, so Winograd can be used to obtain minimal multiplication FFTs and is often used to find efficient algorithms for small factors. Indeed, Winograd showed that the DFT can be computed with only O(N) irrational multiplications, leading to a proven achievable lower bound on the number of multiplications for power-of-two sizes [7].

C. KARATSUBA'S MULTIPLICATION ALGORITHM

Karatsuba's multiplication algorithm uses three single digit multiplications to perform one two-digit multiplication. If Karatsuba's multiplier algorithm is applied recursively, it takes only 3^n single-digit multiplications to multiply a pair of 2^n -digit numbers. This is a significant improvement compared to 4^n single digit multiplications using simple multiplication [8].

The basic step of Karatsuba algorithm can be used to compute the product of two large numbers a and b using three multiplications of smaller numbers, each with about half as many digits as a or b along with some additions and digit shifts. Let a and b represent n-digit strings in some radix R. For any positive integer m, less than n, the two numbers can be divided

as follows: $a = aiR^m + a0$, $b = biR^m + b0$, Where *a0 and b0 are less than* R^m

Product is then $ab = (aiR^m + a0)(biR^m + b0) = u2R^{2m} + u1R^m + u0$ where, $u2 = a1b1$, $u1 = a1b0+a0b1$, $u0 = a0b0$

These formulae require four numbers of multiplications. But, it can be observed that the value of the product ab can be determined using only three numbers of multiplications, at the cost of a few more number of additions in the following manner: after obtaining $u_2 = a_1b_1$ and $u_0 = a_0b_0$ u_1 is calculated as

$u_1 = (a_1 + a_0) (b_1 + b_0) - u_2 - u_0$

Algorithms for Modular p reduction algorithm:

Let k and l be the bit length of p and p [k-2 : 0], respectively.Let x be the input number with a maximum bit length m = 2k.

Input: x; k; l; m; p, Output: y = x mod p.
Step 1: while m > k do
Step 2: x ← x [k − 2: 0] − x [m − 1: k − 1] p [k − 2: 0]
Step 3: m ← m − k + l + 2
Step 4: end while
Step 5: if x < 0 then
Step 6: y ← x + p
Step 7: else
Step 8: y ← x
Step 9: end if
Step 10: return y

D. MODULAR REDUCTION ALGORITHM ANALYSIS

By carefully examining Modular Reduction Algorithm Analysis we conclude that a small value of p [k−2: 0] (small l) reduces the number of loops in the computation. Furthermore, if p [k − 2: 0] is a number with low Hamming weight, the addition of the corresponding segments of x [m − 1: k − 1] can replace the multiplication x [m− 1: k− 1] p [k − 2: 0], which reduces the area usage for modular reduction [10].

III. RESULTS AND DISCUSSIONS

By careful simulation of both karatsuba and wingord algorithm for polynomial multiplication for point wise multiplication it can be concluded that for a complex multiplication, Karatsuba algorithms are faster than Wingord algorithms. Fig 1 give the results of experiments an Intel Core2 Duo CPU T7300 with 4 Gb RAM, set to 800 MHz clock speed [10]. From the graph, it is very clear that wingord algorithm is less slow one than Karatsuba algorithms. Since we ignore the complexity of addition which have complexity of $O(n)$, only concentrate the complexity of

multiplication. Karatsuba have multiplication complexity of O ($n^{1.585}$) [5] where wingord algorithm have a complexity of O($n^{2.585}$) [7]

TABLE 1: Comparison of device utilization and combinational path delay of 88 Karatsuba Multiplier and normal Multiplier.

8 x 8 Multiplier	number of slices	Number of 4 input LUT	Number of Bonded IOBs	Maximum path Delay
Karatsuba	26 out of 2352(1%)	45 out of 4704(0%)	31 out of 140 (22%)	12.338 ns
Normal	38 out of 2352(1%)	73 out of 4704(1%)	32 out of 140 (22%)	15.656 ns

Table 1. shows the comparison of device utilization and combinational path delay of 8x8 Karatsuba Multiplier and normal Multiplier. The number of slices and combinational path delay for 8x8 Karatsuba multiplier are 26 out of 2352 (1%)and 12.338ns respectively. Whereas, the number of slices and combinational path delay for 8x8 normal multiplier are 38 out of 2352 (1%)and 15.656ns respectively [10].

The above observations justify that the proposed Karatsuba multiplier using polynomial multiplication uses less number of slices and at the same time the maximum combinational path delay is also less. Hence, the proposed 8x8 multiplier has speed improvement using lesser space than the corresponding normal 8x8 multiplier [11].

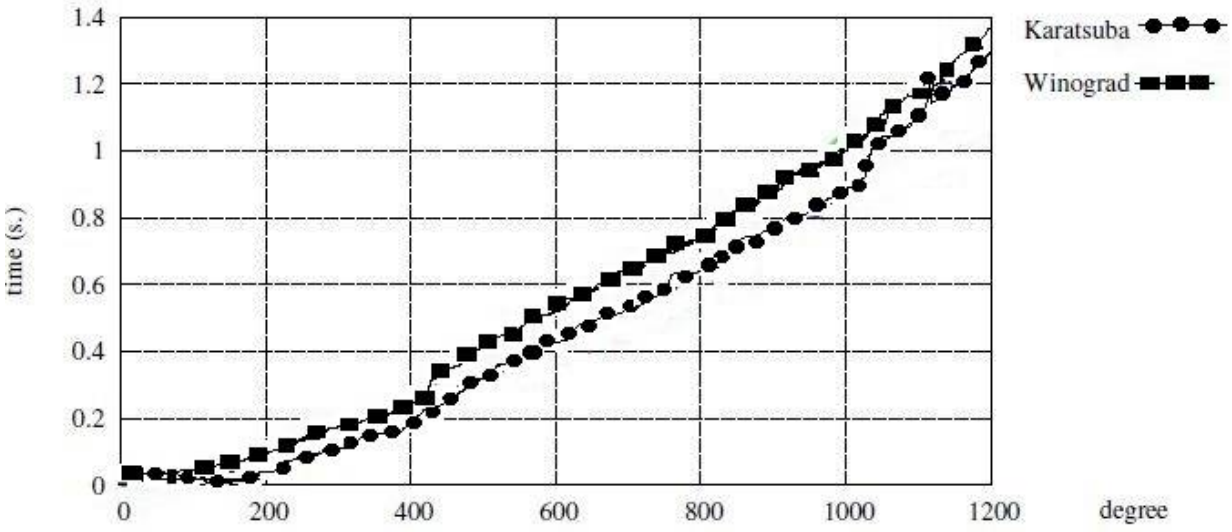

Fig.1: Comparison between Karatsuba and wingord algorithms

TABLE 2: FPGA implementation results of the Modified Montgomery modular multiplier

	Number of clock cycle	Minimum clock period(ns)	Total MMM Latency (μs)	Number of Gates
w=1	160	16.030	2.564	18940
w=4	120	17.895	2.147	32564
w=8	60	19.798	1.187	37017
w=16	30	23.162	0.694	49112
w=32	15	62.560	.938	72690

The operation speed of polynomial multiplication can be further reduced using Modular reduction algorithm in which word size are reduced, hence total multiplication is reduced. Table 2 shows the FPGA implementation result using Xilinx ISE 14.7 simulator.

When the word length w increases, the number of clock cycles for completing one MMM decreases, but the minimal clock period increases. The total MMM latency reaches an optimum for w = 16. The circuit becomes larger when w increases.

IV. CONCLUSION

Polynomial multiplication is the most intensive and time consuming operation in the modern cryptography. So, we need to maximize the efficiency in fast computation and reduce area and cost. In this Paper, we compare both wingord and karastuba algorithm which are based on FFT multiplication. It is very clear that Karastuba algorithm is more

efficient in computation speed. Also, have area-cost efficiency than wingord polynomial algorithm. Modular reduction like modified Montgomery is further improved the efficiency of FFT multiplier.

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 21, no. 2, pp. 120126, 1978.
[2] Donald Donglong Chen, Nele Mentens, Frederik Vercauteren, Sujoy Sinha Roy, Ray C.C. Cheung, Derek Pao, and Ingrid Verbauwhede, "High-speed Polynomial Multiplication Architecture for Ring-LWE and SHE Cryptosystems," IEEE Transactions on Circuits and Systems, 2015 ,Volume 1,pp.157-166.
[3] Xin Xiao, Erdal Oruklu,, "An Efficient FFT Engine With Reduced Addressing Logic," IEEE transactions on circuits and systemsii, vol. 55, no. 11, November 2008.
[4] Chaohui Du and Guoqiang Bai, "A Family of Scalable Polynomial Multiplier Architectures for Ring-LWE Based Cryptosystems," 2015 IEEE Trustcom/BigDataSE/ISPA , ,2015, Volume: 1 Pages: 392 - 399.
[5] J. von zur Gathen and J. Shokrollahi, " Efficient FPGA-based Karatsuba multipliers for polynomials over F2. Selected Areas in Cryptography, "Lecture Notes in Computer Science, 3897, pp. 359369
[6] Ling Ding1 and Eric Schost,"Code Generation for Polynomial Multiplication", Computer Algebra in Scientific Computing, Volume 5743 of the series Lecture Notes in Computer Science pp 66-78
[7] G.A.Sathishkumar and Dr.K.Boopathy bagan"A Hardware implementation of Winograd Fourier Transform algorithm for Cryptography"Ubiquitous Computing and Communication Journal
[8] Zhou, Gang; Michalik,Harald; Hinsenkamp, Lszl " Complexity Analysis and Efficient Implementations of Bit Parallel Finite Field Multipliers Based on Karatsuba-Ofman Algorithm on FPGAs.", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18 (7), pp.1057-1066.
[9] Lima, Juliano B., et al., "A Karatsuba-Based Algorithm for Polynomial Multiplication in Chebyshev Form. "IEEE Transactions on Computers, 59(6), pp. 835-841.
[10] M.-D. Shieh, J.-H. Chen, W.-C. Lin, and H.-H. Wu, "A new algorithm for high-speed modular multiplication design", Circuits and Systems I: Regular Papers,IEEE Transactions on ,vol. 56, no. 9, pp. 20092019, Sept 2009.
[11] A. Karatsuba and Y. Ofman, "Multiplication of multidigit numbers on automata,"Soviet Physics Doklady, vol. 7, no. 7, pp. 595596, 1963.