International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
www.ijset.com, Volume No.1, Issue No.3, pg : 180-183 01 July 2012

New Generation Fast and Optimized Modulo (2^n +1) Multiplier for IDEA

Pravinkumartiwari[1] ,Momd.abdullah[2], Rajesh nema[3]

[1,2,3] Department of Electronics & Communication engg.
NRI Institute of Information Science & Technology Bhopal, INDIA
pravin.mtech11@gmail.com[1],mab434@gmail[2],rajeshnema2010@rediffmail.com[3]

Abstract **: International data encryption algorithm, a federal information processing standard is an approved cryptographic algorithm that can be used to protect electronic data. This paper present the IDEA algorithm with regard to FPGA and the very high speed integrated circuit hardware description language. Synthesizing and implementation of the VHDL code carried out on Xilinx-project navigator, ISE suite. In this paper , an efficient hardware design of the IDEA using modulo (2^n+1) multiplier as the basic module proposed for faster, smaller and low power IDEA hardware circuit. Experimental measurement result show that the proposed design is faster and smaller and also consume less power than similar hardware implementation making it a viable option for efficient This paper talks of IDEA 64 bit plain text, 128 bit key and 64 bit cipher text.**

Keywords:-IDEA, cryptographic algorithm, Xilinx, FPGA, modulo 2^n+1 multiplier

I. INTRODUCTION

The importance of cryptography applied to security in electronic data transactions has acquired an essential relevance during the last five years. Each day millions of users generate and interchange large volumes of information in various fields, such as financial and legal files, medical reports and bank services via Internet, telephone conversations, and e-commerce transactions. These and other examples of application deserve a special treatment from the security point of view, not only in transport of such information but also in its storage. In this sense, cryptography techniques are especially applicable. This implementation will be useful in wireless security like military communication and mobile telephone where there is a greater emphasis on the speed of communication .In this paper, the cipherused is a symmetric key block cipher .It takes its input as 64 bit plain text and gives a 64 bit cipher text as output using a 128 bit key. While working on plain text, it divides the input data in to 16 bit sub-blocks and operates on each block. It is described as one of the more secure block algorithm due to its high immunity to attacks. In spite of the fact thatData Encryption standard (DES) is another popular symmetric block cipher which is used in several financial and business application and its drawback is the short key word length .Moreover unlike DES,IDEA doesn't need any S-box or P-box is required for implementing this cipher. The most crucial module part of this algorithm is the design of the multiplier modulo a Fermat prime, which is one of the algebraic group operation used and entire speed of IDEA depends on this module. So designing the multiplier is a major during the hardware or software implementation of IDEA because its speed is a big issue when hardware implemented IDEA is used in real time application. The overall objective for hardware implementation of IDEA is to minimize the hardware requirements which result in efficient use of silicon area and at the same time improve the processing speed and high throughput of data. As the performance of IDEA cipher depends entirely on the modulo(2^n+1) multiplier design, the main objective is to design an efficient andfast modulo multiplier which is to be used in the entire IDEA algorithm.

The paper is organized as follows; section II describe the IDEA algorithm. Section III describe the modulo (2^n+1)multiplier. SectionIV describe the proposed multiplier. SectionVdiscuss the results and comparisons with previous schemes. Conclusion and references are given in section VI and VII respectively.

II. THE IDEA ALGORITHM

In this section, the entire algorithm for the IDEA block cipher is elaborated. It is a symmetric key cipher. The block size of data on which IDEA operates, is of 64 bit and the key size is of 128 bits. But all data operations in IDEA cipher are in 16 bit unsigned integers. The length of the incoming data should be either in normal in integer multiple of 64 bits or if not, is made by using

.

IDEA is based on mixing operation of three different algebraic groups which are

- XOR (bitwise).
- Addition modulo 2^n.
- Multiplication modulo (2^n + 1).

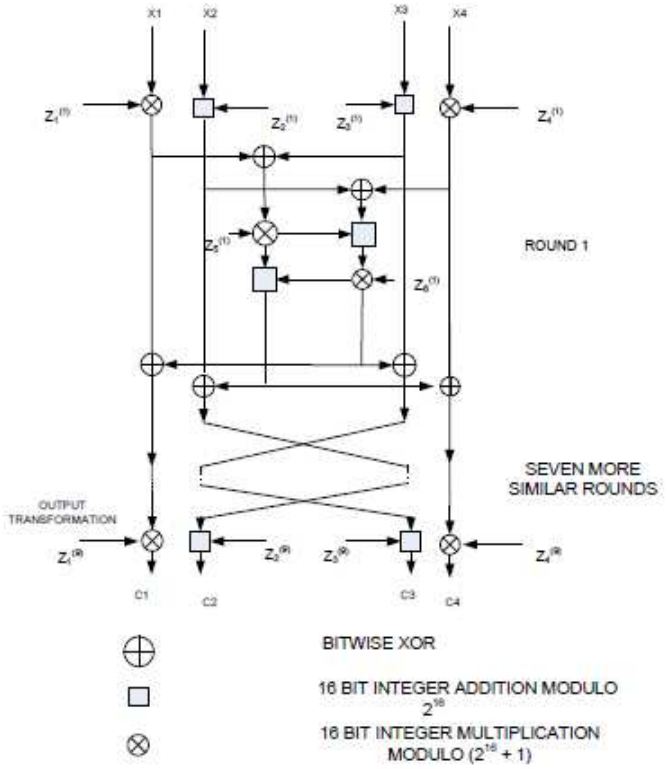


Figure1. Basic structure of IDEA Cipher and its data flow

The security of IDEA depends on these three operations. The basic structure of IDEA cipher is shown in Figure .

The IDEA cipher consists of 8 rounds which are identical in nature and a last output transformation round which is similar to upper half of any round. Before the starting of 1st round, the input 64 bit plain text is divided into four 16 bit sub-blocks, X1, X2, X3 and X4 respectively. At the end of encryption phase, four 16 bit sub-blocks of cipher text is created. Each round uses six 16 bit sub-key blocks Z1 $^{(n)}$, Z2 $^{(n)}$,… , Z6$^{(n)}$.which are made from the input 128 bit key. The super-script n denotes the nth round. The output transformation phase, which is considered as 9th or the last round, uses 4 sub-keys, Z1 $^{(9)}$, Z1 $^{(9)}$, Z1 $^{(9)}$, Z1 $^{(9)}$. Every round except the 1st round uses the output sub-blocks produced in the previous round. In between every round, the 2nd and the 3rd sub-blocks are swapped. The entire algorithm uses only three different algebraic group operations which are XOR, addition modulo 2^n and multiplication modulo (2^n+1). The encryption phase of IDEA thus uses [(8*6) + 4] i.e. 52 sub-key blocks, which are made from the 128 bit input key. As IDEA involves only algebraic operations, no look-up tables or S-Boxes are used like DES or AES.

The decryption phase of IDEA is identical to that of the encryption phase. It uses the same sequence of operations as in the encryption phase. The only change is that the sub-keys are reversed and are slightly different. That means the sub-keys which are used in round 1 during encryption phase are manipulated during last round of decryption phase. The subkeys used in decryption are either additive or multiplicative inverse of the sub-keys used in the encryption phase.

III. MODULO (2^n+1) MULTIPLIER

Modulo(2^n+1)multiplier is one of the critical component applications in the area of digital signal processing, data encryption and residue arithmetic that demand high-speed and low-power operation. In this paper, efficient hardware architecture of modulo ($2n$+ 1) multiplier is proposed and validated to address the demand. The proposed modulo ($2n$+1) multiplier has three major functional modules including partial products generation module, partial products reduction module and final stage addition module. Modulo arithmetic has been widely used in various application such as digital signal processing where the residue arithmetic is used for digital filter design. Also, the number of wireless and internet communication nodes has grown rapidly. The confidentiality and the security of the data transmitted over these channels has becoming increasingly important. Cryptographic algorithms like International Data Encryption Algorithm (IDEA) is frequently used for secured transmission of data. Modulo $2n$ addition and modulo (2^n +1) multiplication are the crucial operations in the IDEA algorithm and also modulo (2^n + 1) arithmetic operations are used in Ferment number transform computation. Now a day, modulo arithmetic is frequently used in fault tolerant design of ad-hoc networks, digital and linear convolution architectures. Apart from these, residue arithmetic is extremely efficient for image processing, speech processing and transforms all of which are extremely important in today's high dense computing world.

IV. PROPOSED MULTIPLIER

There are several multipliers existing for idea. Here we are presenting a new kind of modulo multiplier which is highly optimized as compared to previous one. The description for proposed multiplier is as follows. In general if we multiply the two n bit number ,then we get 2n bit number. Store this 2n bit number(result) in to temporary register ie.t1,then make the length of modulo(2^n+1) equal to the length of 2n bit number ie.t1, by consented zeroes('0's) after the LSB bit of the modulo(2^n+1) and store this value into t2. Here we defining the subsequent steps by block/flow diagram, which is easy to understand. Here we take n=16…

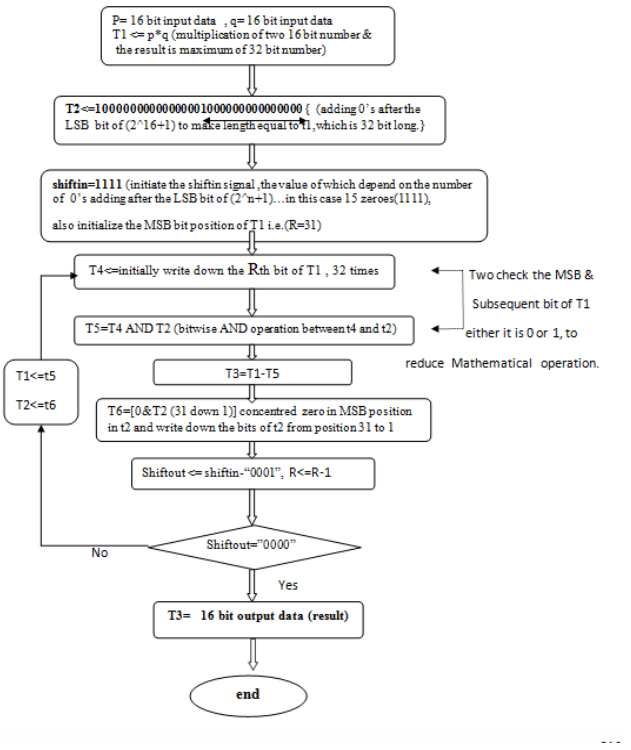


Figure 2.Block diagram/flow diagram of proposed multiplier

Let we understand This proposed multiplier using an example for n=4 bit .the result (worst case) of multiplication of two 4 bit number is $11111111(255)_{10}$ if we calculate the mod 2^n+1 (n=4)of this number then we get the result is 00000000 $(0)_{10}$.

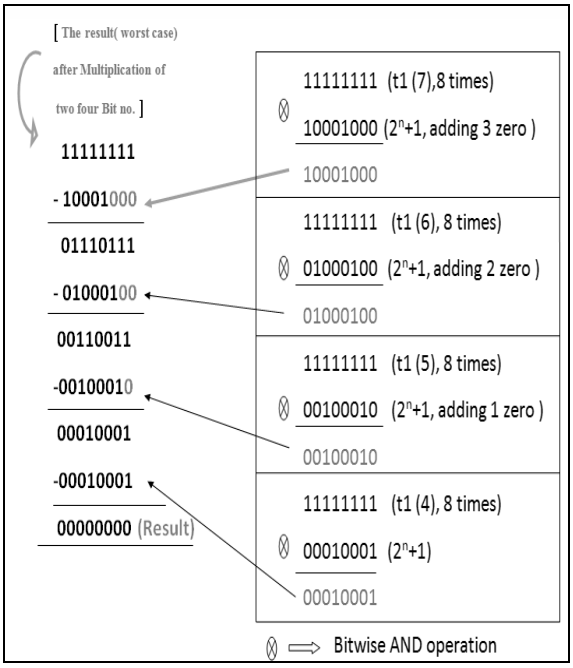


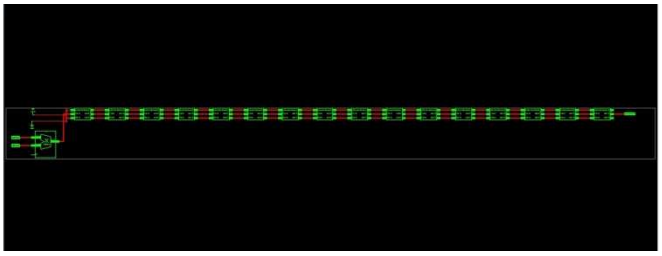
Figure 3. example of two 4 bit number

V. RESULT

The parameter used to evaluate the quality of the modulo multiplier are slices, internal multiplier used and throughput per slices(TPS). Xilinx synthesis tools measure the amount of used resources in terms of Configurable Logic Block (CLB's) or slices, where one CLB is ebullient to four slices. The principal difference between the reportsfrom different manufacture is the basic element definition and its interconnection with others of the same kind. For this reason, theresults of the synthesis are compared with other implementations that have been targeted on chips from the same manufacturer.

TABLE 1: **Synthesis result . DeviceVirtex2P-XC2VP40-FG676-7**

PARAMETERS	Used	Available	Utilization
Number of 4 input LUT's	376	93,184	1%
Number of slices	192	46,592	1%
Number of bonded IOB's	48	824	5%
Number of mul 18*18s	1	168	1%



The results of the implimantation in terms of area & SPEED are summarized in table 1.table 2 represent the results obtained with other hardware implimantation.

Table 2.**Performance result for comparison**

Preferred FPGA device	Multiplier [1]	Multiplier [2]	Proposed multiplier
Number of internal Multiplier used	4	3	1
Number of slices(LUT)used	264	-	192
Maximum clock frequency	10.9MHz	8.25 MHz	18.12MHz
Size of bits processed	16	16	16
Throughput	703.36Mbps	-	1159.68Mbps

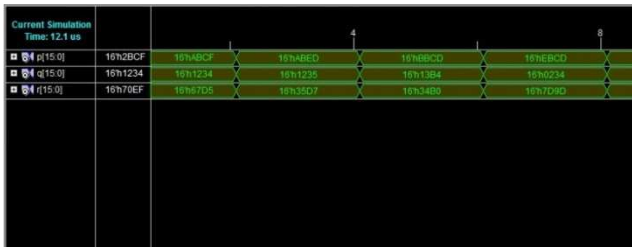


Figure 4. Simulation window of implimentied multiplier

Figure 5. RTL view of proposed multiplier

VI. CONCLUSION

This paper presents a high speed, low area, cost effective idea cipher for encryption using a basic 64-bit iterative architecture, targeted towards the Spartan family of FPGAs. This architecture is based on previous work on the cipher design. In this work a modulo (2n+1) multiplier is modified. The number of clock cycle required to encrypt a single block has been reduced and amount of hardware resources has been optimized. The architecture needs fewer logic cells than other cipher and uses as few memory blocks as possible.

REFERENCES

[1] Sourav Mukherjee and BibhudattaSahoo, "A Hardware implementation of IDEA cryptosystem using a recursive multiplication approach.", International Conference on Electronic Systems (ICES-2011), pp 383 - 389, 2011,

[2] YI-JUNG CHEN1, DYI-RONG DUH2 AND YUNGHSIANG SAM HAN, "Improved Modulo (2n + 1) Multiplier for IDEA", journal of information science and engineering 23, 907-919 (2007).

[3] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W. Fichtner,"A 177mb/s VLSI implementation of the international data encryption algorithm,"IEEE Journal of Solid-State Circuits, Vol. 29, 1994, pp. 303-307.

[4] Modugu.R, Yong-Bin Kim, MinsuChoi,"Design and performance measurement of efficient IDEA crypto-hardware using novel modular arithmetic components", Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE, 3-6 May2010,pp1222-1227.

[5] Rahul Ranjan and I. Poonguzhali, "VLSI Implementation of IDEA Encryption Algorithm", Mobile and Pervasive Computing (CoMPC–2008).

[6] SomayehTimarchi, KeivanNavi, "Improved Modulo 2^n+1 Adder Design", International Journal of Computer and Information Engineering 2:7 2008.

[7] X.Lai and J.L Massey "A Proposal for a New Block Encryption Standard," in advances in CrypEUROCRYPT 90,Berlia,Germany: Springer Verlag pp. 389-404, 1990.

[8] AnttiH¨am¨al¨ainen, MattiTommiska, and JormaSkytt¨, "6.78 Gigabits per Second Implementation of the IDEA Cryptographic Algorithm",2002 Springer-Verlag, pages 760-769.

[9] M.P. Leong, O.Y.H. Cheung, K.H.Tsoi and P.H.W.Leong "ABit Serial Implementation of the International Data Encryption Algorithm IDEA" ©IEEE 2000.

[10] P. Kitsos, N. Sklavos, M.D. Galanis, O. Koufopavlou , "64 Bit Block

[11] ciphers: Hardware Implementations and Comparison analysis",593-604,3rd November,2004,Elsevier

[12] Thaduri,M.,Yoo,S. and Gaede,R, " An Efficient Implementation ofIDEA encryption algorithm using VHDL", ©2004 Elsevier.

[13] Allen Michalski1, Kris Gaj, Tarek El-Ghazawi, "An Implementation Comparison of an IDEA Encryption Cryptosystemon Two General-Purpose Reconfigurable Computers"

[14] SarangDharmapurikar and John Lockwood, "Fast and Scalable Pattern Matching for Network Intrusion Detection Systems"IEEE Journal on Selected Areas in Communications: Oct. 2006, Volume: 24, pp. 1781- 1792 .

[15] Chiranth E, Chakravarthy H.V.A, Naga mohanareddy P, Umesh T.H, Chethan Kumar M., "Implementation of RSA Cryptosystem Using Verilog" International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011.