VHDL Implementation of IDEA Architectures

A. D. Chaudhari, S. D. Shirbahadurkar

Abstract* – Cryptography is the art of keeping data secure from unauthorized access so as to guarantee that only the intended users can access it. Data security is an important issue in computer networks and cryptographic algorithms are essential parts in network security.This paper covers the implementation of the International Data Encryption Algorithm (IDEA) using Very Large Scale Integrated Circuits Hardware Description Language (VHDL) with the help of Xilinx – ISE 9.1. In terms of security, this algorithm is very much superior. In IDEA, the plaintext and the cipher text are 64 bit blocks, while the secret key is 128 bit long. The cipher is based on the design concept of mixing operations from different algebraic groups.

***Keywords*-Cryptographic Algorithm, IDEA, Modulo Multiplier, VHDL, Xilinx.**

I. INTRODUCTION

The confidentiality and security are important for the data to be transmitted through a communication system. The Internet today is a widespread information infrastructure, but it is also an insecure channel for sending messages. Thus there is a need for encrypting the data before it is transmitted through any medium. International Data Encryption Algorithm (IDEA) have found various applications in secure transmission of the data in networked instrumentation and distributed measurement systems. IDEA provides data integrity, authentication, and confidentiality. Previously there are many encryption standards for secure data transmission like RC5, RC6,DES and AES. But, IDEA is a superior encryption standard compared to any other encryption techniques.

Cryptography:

Encryption is the conversion of information, the plain text or message, into code, the cipher which is intelligible only for an authorized receiver. Data encryption is applied to ensure secrecy of a message transferred. Historically, cryptographic techniques have been developed for diplomatic or military applications; today they can be found everywhere in private and public sectors where confidential information has to be processed, transferred, and stored. An important characteristic of modern cryptography is the use of publicly known, i.e. published, algorithms. The secrecy is therefore not kept in algorithm itself, but only in a small additional piece of information shared between sender and receiver the key.IDEA is a block cipher designed by Xuejia Lai and James L. Massey. It is a minor revision of an earlier cipher, proposed encryption standard (PES).IDEA is a block cipher that uses a 128-bit key to encrypt 64bit data blocks. The 52 subkeys are all generated from the 128-bit original key. IDEA algorithm uses 52, 16bit key sub-blocks, i.e. six subkeys for each of the first eight rounds and four more for the ninth round of output transformation(8*6+4=52).

IDEA is designed in such a way that it facilitates both software and hardware implementations. The hardware implementation is designed to achieve High speed. The advantage of software implementation are flexibility & low cost. The structure of the cipher must be regular and modular, in order to make VLSI implementation simpler. The efficiency of the IDEA cipher can be improved if efficient basic modules such as modulo multipliers and adders are used. In IDEA, the plaintext is 64 bits in length and the key size is 128 bits long. The design methodology behind the IDEA algorithm is based on mixing three different operations XOR, addition and multiplication.

II. IDEA ENCRYPTION/DECRYPTION

In general terms, there are two types of keys based encryption algorithms: symmetric and public key. Symmetric algorithms in which the encryption key can be calculated from the decryption key and vice-a-versa while in most of them both (encryption and decryption) keys are identical .symmetric algorithms can be further divided in two categories: The first category includes the ones that operate on the plaintext a single bit at a time and they are called stream algorithms or stream ciphers. The other category includes these algorithms which operate on the plaintext in groups of bits and the algorithms are called block algorithms or block ciphers. IDEA is a secret-key cipher whose encryption and decryption processes are symmetric. The cipher IDEA is an iterated cipher consisting of 8 rounds followed by an output transformation. It takes 64bit plaintext inputs and produces 64bit cipher text outputs by using a 128bit key.

a) General Architecture

The functional representation of the encryption process is shown in fig.1. The process consists of eight identical encryption steps followed by an output transformation. The structure of the first round is shown in detail.

The eight rounds are performed using the combination of three algebraic operations:

⊕ Bitwise XOR,

⊞ Addition modulo 2^{16}, and

⊙ Modified multiplicationmodulo2^{16}+1

In each round of the 8 rounds of algorithm, the following sequence of events are performed:

1. Multiply X1 by the first subkey.
2. Add X2 and the second subkey.
3. Add X3 and the third subkey.
4. Multiply X4 by the fourth subkey.
5. XOR the results of Steps 1 and 3.
6. XOR the results of Steps 2 and 4.
7. Multiply the results of Step 5 by the fifth sub key.

Revised Manuscript Received on June 06, 2013.

Ms.A.D.Chaudhari, Padmabhooshan Vasantdada Patil Institute of Technology, E&TC Department, Bavdhan, Pune, India.

Dr.S.D.Shirbahadurkar, Padmabhooshan Vasantdada Patil Institute of Technology, E&TC Department, Bavdhan, Pune, India.

8. Add the results of Steps 6 and 7.
9. Multiply the results of Step 8 by the sixth subkey.
10. Add the results of Step 7 and 9.
11. XOR the results of Steps 1 and 9.
12. XOR the results of Steps 3 and 9.
13. XOR the results of Steps 2 and 10.
14. XOR the results of Steps 4 and 10.

The computational process used for decryption of the cipher text is essentially the same as that used for encryption of the plaintext. The only difference compared with encryption is that during decryption, different 16-bit key sub-blocks are generated.

More precisely, each of the 52 keys, 16-bit key sub-blocks used for decryption is the inverse of the key sub-block used during encryption in respect of the applied algebraic group operation. Additionally, the key sub-blocks must be used in the reverse order during decryption in order to reverse the encryption process as shown in Table I.

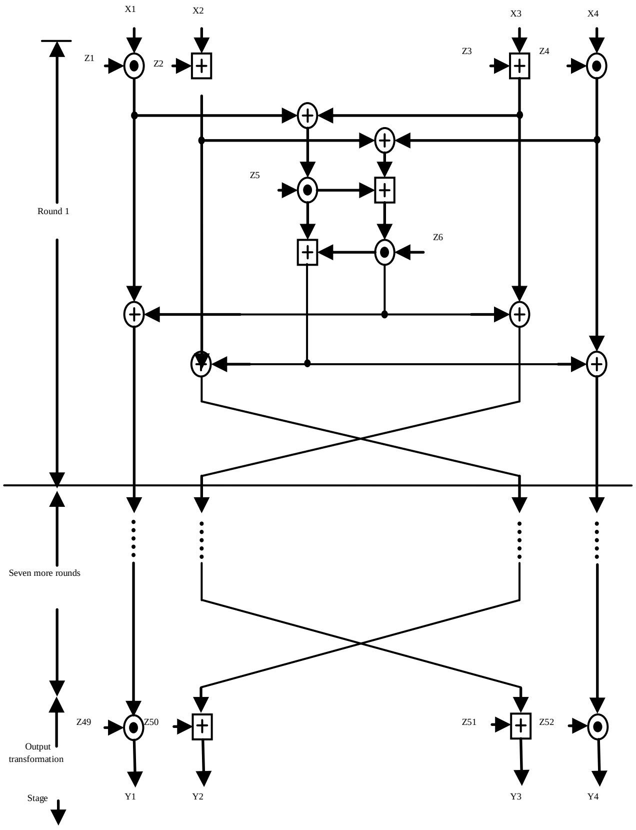

Fig. 1 IDEA Architecture

Key Scheduling

Different 16-bit sub-blocks have to be generated from the 128- bit key. The 52, 16-bit key sub-blocks which are generated from the 128-bit key are produced as follows:

- First, the 128-bit key is partitioned into eight 16-bit sub-blocks which are then directly used as the first eight key sub-blocks.
- The 128-bit key is then cyclically shifted to the left by 25 positions, after which the resulting 128-bit block is again partitioned into eight 16-bit sub-blocks to be directly used as the next eight key sub-blocks.
- The cyclic shift procedure described above is repeated until all of the required 52, 16-bit key sub-blocks have been generated.

The key sub-blocks used for the encryption and the decryption in the individual rounds are shown in below Table.

TABLE I:
IDEA ENCRYPTION AND DECRYPTION SUBKEYS

Round	**Encryption Subkeys**	**Decryption Subkeys**
1	$Z_1Z_2Z_3Z_4Z_5Z_6$	$Z^{-1}{}_{49}–Z_{50}–Z_{51}Z^{-1}{}_{52}Z_{47}Z_{48}$
2	$Z_7Z_8Z_9Z_{10}Z_{11}\ Z_{12}$	$Z^{-1}{}_{43}–Z_{45}–Z_{44}Z^{-1}{}_{46}Z_{41}Z_{42}$
3	$Z_{13}Z_{14}Z_{15}Z_{16}Z_{17}Z_{18}$	$Z^{-1}{}_{37}–Z_{39}–Z_{38}Z^{-1}{}_{40}Z_{35}Z_{36}$
4	$Z_{19}Z_{20}Z_{21}Z_{22}Z_{23}Z_{24}$	$Z^{-1}{}_{31}–Z_{33}–Z_{32}Z^{-1}{}_{34}Z_{29}Z_{30}$
5	$Z_{25}Z_{26}Z_{27}Z_{28}Z_{29}Z_{30}$	$Z^{-1}{}_{25}–Z_{27}–Z_{26}Z^{-1}{}_{28}Z_{23}Z_{24}$
6	$Z_{31}Z_{32}Z_{33}Z_{34}Z_{35}Z_{36}$	$Z^{-1}{}_{19}–Z_{21}–Z_{20}Z^{-1}{}_{22}Z_{17}Z_{18}$
7	$Z_{37}Z_{38}Z_{39}Z_{40}Z_{41}Z_{42}$	$Z^{-1}{}_{13}–Z_{15}–Z_{14}Z^{-1}{}_{16}Z_{11}Z_{12}$
8	$Z_{43}Z_{44}Z_{45}Z_{46}Z_{47}Z_{48}$	$Z^{-1}{}_{7}–Z_{9}–Z_{8}Z^{-1}{}_{10}Z_{5}Z_{6}$
9	$Z_{49}Z_{50}Z_{51}Z_{52}$	$Z^{-1}{}_{1}–Z_{2}–Z_{3}Z^{-1}{}_{4}$

b) Pipelined Architecture

Fig.2 shows the pipelined architecture of IDEA algorithm. In pipeline architectures, registers are provided at different stages of the algorithm. At each clock cycle, the output of a stage is shifted to the next stage. Thus, at the first clock cycle one input block should be loaded. At the next clock cycle, a second block must be loaded and so on. Once the pipeline is filled, then an output value will be ready at each clock cycle.

Pipeline is a fast approach but cost has to be paid in terms of hardware resources. Unfortunately, that approach cannot be fully utilized for IDEA algorithm computation due to the inherent dependencies. As it was explained, the second iteration cannot be started until the computations for first iteration have been completed. However a sort of pipelining can be achieved for different operations of the similar stage.

c) Serial Architecture

In serial architecture, 64 bit data is given as input to data block. Instead of using 8 general rounds, we are using only one general round in this serial architecture. By applying reset & clock pulses to clock generator, it will generate different signals (sub keys & clock signals) to different blocks in architecture.

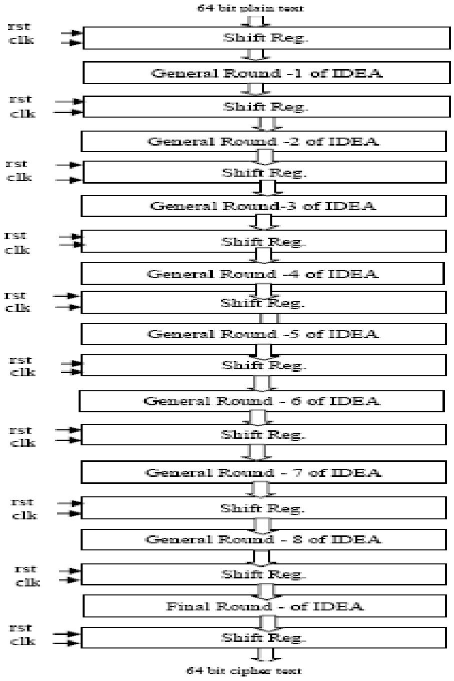

Fig. 2 IDEA Pipelined Architecture

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

III.RESULTS

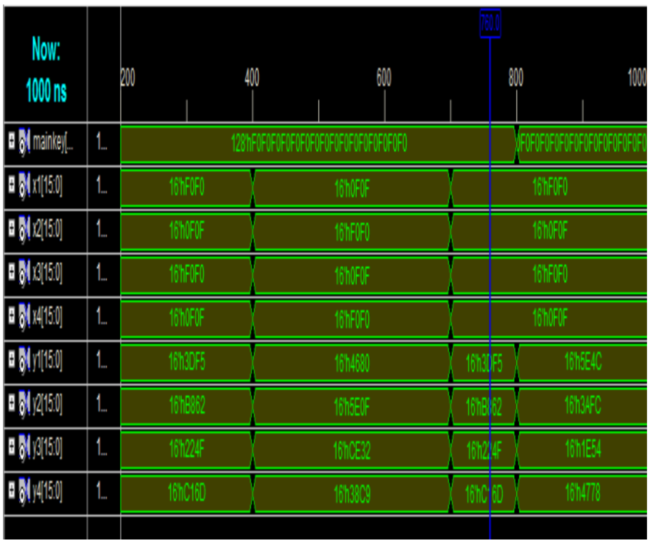

Fig. 3 Simulation results of IDEA-General

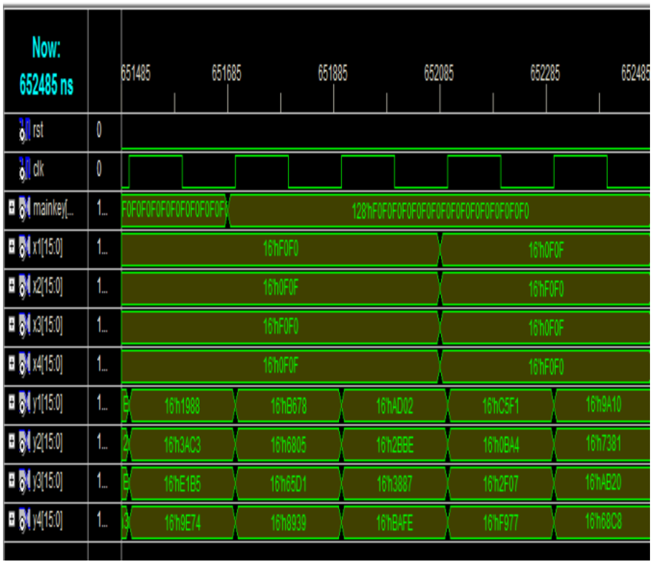

Fig. 4 Simulation results of IDEA-Pipelined

IV. CONCLUSION

In this paper the implementation of IDEA based on modular arithmetic components. The general and pipelined architectures for IDEA algorithm simulated in Xilinx. The choice of IDEA as the encryption/decryption algorithm ensures the strength of the data encryption algorithm. This can be used in very high speed & low power encryption/ decryption algorithms.

REFERENCES

1. Modugu.R, Yong-Bin Kim, Minsu Choi, "Design and performance measurement of efficient IDEA cryptohardware using novel modulararithmetic components", Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE, 3-6 May2010, pp1222-1227.
2. O.Y.H. Cheung, K.H.Tsoi, P.H.W. Leong, and M.P Leong. *"Tradeoffs in Parallel and Serial Implementations of the International Data Encryption Algorithm IDEA"*
3. M.P.Leong, O.Y.H Cheung, K.H.Tsoi, and P.H.W.Leong *"A Bit-serial implementation of the International DataEncryption Algorithm IDEA"*. In B.Hutchings,editor, IEEE Symposium of Field Programmable Custom Computing Machines, pages 122 -13. IEEE Computer Society, 2000.
4. Efficient Online Self-Checking Modulo 2^n+1 Multiplier Design "IEEE TRANSACTIONS ON COMPUTERS,VOL.60, NO.9, SEPTEMBER 2011" Wonhak Hong, RajashekharModugu, and Minsu Choi, Senior Member, IEEE
5. Thaduri, M., Yoo, S.M. and Gaede, R., "An efficient implementation of IDEA encryption algorithm using VHDL", ©2004 Elsevier.
6. R. Modugu, N. Park and M. Choi, A Fast Low-Power Modulo 2^n+1 Multiplier Design, 2009 IEEE International Instrumentation and Measurement Technology Conference, pp.951-956, May 2009.
7. Rahul Ranjan and I. Poonguzhali, "VLSI Implementation of IDEA Encryption Algorithm", Mobile and Pervasive Computing(CoMPC – 2008).
8. Dr. Salah Elagooz, Dr. Hamdy,Dr. KhaledShehata and Eng.M. Helmy, "Design and implementation of Highand Low Modulo (2^{16}+1) multiplier used in IDEA Algorithm on FPGA", 20th National Radio Science Conference CAIRO, Egypt , March 18-20 , 2003.
9. Zimmermann, A.Curiger, H.Bonnenberg,H.Kaeslin,N.Felber and W.Fichtner, A 177 Mb/s VLSI implementation of the international data encryption algorithm, IEEE J.Solid-State Circuits, 1994, 29, (3), pp. 303-307