ISSN 2348 – 8034
Impact Factor- 3.155

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

FLEXIBLE AND ENERGY - EFFICIENT RECONFIGURABLE HARDWARE FOR ZUC STREAM CHIPER

Sachin S. Chaudhari*[1] and Prof. Sanjay S. Badhe[2]
[*1] Student, E & TC Engineering, Dr.D.Y.Patil College Of Engg., Ambi, Maharashtra, India.
[2] Asst. Prof., E &TC Engineering, Dr.D.Y.Patil College Of Engg., Ambi, Maharashtra, India.

ABSTRACT

In the world of cryptography, stream ciphers are recognized as primitives used to confirm secrecy over a communication channel. One public way to form a stream cipher is to use a key stream generator to yield a pseudorandom sequence of symbols. ZUC is a stream cipher that create the heart of the 3GPP confidentiality algorithm 128-EEA3 and the 3GPP integrity algorithm 128-EIA3, proposing consistent security services in Long Term Evolution networks (LTE), which is a contestant standard for the 4G network. A completed hardware application is presented in order to spread accurate performance outcomes in LTE systems. Stream ciphers are further efficient when executed in hardware atmosphere, like Field Programmable Gate Array (FPGA). The project is coded using VHDL language also for the hardware execution, a XILINX Virtex-5 FPGA is used. In this notepaper a reconfigurable implementation of ZUC stream cipher with the help of Carry Look Ahead Adder is offered. This accomplished a throughput of 3.3180 Gbps.

Keyword *: - Long Term Evolution, web security, ZUC,4G, FPGA, Virtex 5.*

I. INTRODUCTION

For encryption devotions there occur, mostly, two kinds of primitives, block and stream ciphers. Block ciphers are standard primitives that have been studied for few years. Composed project methods and cryptanalysis of block ciphers permitted to progress such a standard for encryption as Rijndael (AES). This cipher is broadly accepted, and it has robust opposition beside various types of doses. On the other side, even though the idea of stream ciphers seemed lengthy ago, the open study and examination of these primitives initiated only about 20 years before. It is generally alleged that stream ciphers can be slighter and much more rapidly than block ciphers when executed. Unluckily, we still do not have plenty knowledge about the project and cryptanalysis of stream ciphers.

II. LONG TERM EVOLUTION

Today there are various stream cipher algorithms projected in both educational and industrial study. In the arena of telecommunications, the world is moving into 4th Generation (4G) standard. During the previous few years, the 3rd Generation Partnership Project (3GPP) has succumbed Long Term Evaluation Advanced (LTE-Advanced), which is the improvement of the LTE standard, as a candidate for the 4G network. Long Term Evolution (LTE), is the next-generation network outside 3G that allow fixed to movable migrations of Internet applications such as Voice over IP (VoIP), Video streaming, Music downloading, Mobile TV and various others. LTE networks will also offer the capability to support an explosion in request for connectivity from a innovative generation of consumer devices tailored to those new mobile applications.

The present radio interface protection algorithms for LTE, 128-EEA1 for confidentiality and 128-EIA1 for integrity have been made by SAGE/ETSI Security Algorithms Group of Experts. 128-EEA1 and 128-EIA1 are built on SNOW3G stream cipher. Also, the 3rd Generation Partnership Project (3GPP), organized with the GSM Association agrees a second set of algorithms, 128-EEA2 and 128-EIA2, which are made on AES block cipher. Lastly, 3GPP with GSM association postulates a third set of algorithms for confidentiality and integrity the 128-EEA3 and 128-EIA3 correspondingly. Both ciphers are based upon ZUC stream cipher. The most critical reason for these new ciphers is that LTE will be used in various countries. But Chinese rules will not permit those algorithms to be used in China, because they were not made in China. However, ZUC has been constructed in China, and thus that it can be used in China. In this job, an effective FPGA implementation of ZUC stream cipher is offered. The benefits of Virtex- 5 FPGA are described with the help of embedded functions like Digital Signal Processing (DSP) blocks, with the aim to reduce the registers and Look-Up Tables in the plan.

III. BLOCK DIAGRAM

An FPGA Employment of ZUC Stream Cipher made of private computer, Xilinx Virtex-5 FPGA kit consisting ZUC Algorithm, Power Supply, Data to be Encrypted and encrypted data. ZUC encryption will be directed from PC to Xilinx kit. Xilinx kit will encrypt data with encryption key and encrypted data will again directed back to PC.

ISSN 2348 – 8034
Impact Factor- 3.155

On PC using Hyper-Terminal, we can see data before encryption ZUC encryption key and encrypted data. Xilinx® Field Programmable Gate Arrays (FPGAs) are greatly flexible, reprogrammable logic devices that influence progressive CMOS manufacturing technologies, alike to other industry-leading processors and processor peripherals.

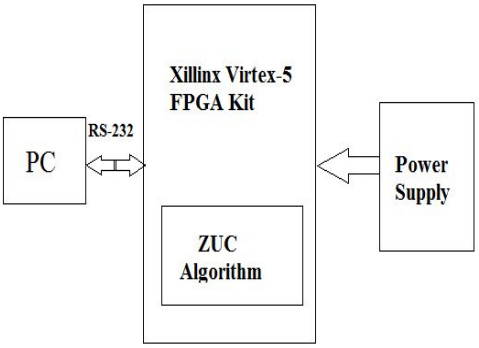

Fig.1 Block Diagram of System

Xilinx FPGAs are wholly user programmable. For FPGAs, the program is called a configuration bit stream, which states the FPGA's functionality. The bit stream loads into the FPGA at system power-up. The process whereby the describing data is loaded or programmed into the FPGA is called Configuration. Configuration is planned to be elastic to lodge different application wishes and wherever possible, to impact present system resources to reduce system costs. Xilinx FPGAs optionally load or boot themselves automatically from an external nonvolatile memory.

IV. ZUC STREAM CIPHER

Cipher systems are usually classified as block ciphers and stream ciphers. Block ciphers tend to concurrently encrypt sets of characters, whereas stream ciphers activate on individual characters of a plain text message one at a time. ZUC is a word-oriented stream cipher, which is the main function of 3GPP confidentiality algorithm: 128-EEA3 and the 3GPP integrity algorithm: 128-EIA3.It accepts a 128-bit Key along with a 128-bit Initial Vector (IV) as input, and outputs a keystream of 32-bit words. The operation of ZUC has two parts: key initialization part and working part. In the first part, a key initialization on LFSR (Linear Feedback Shift Register) is done. The second part is a working stage. In this part LFSR does not receive any input. After working part, during the key stream creating, with every clock tick, it creates a 32-bit word of output. In the specification, the algorithm is break into three logical layers: a linear feedback shift register (LFSR) of 16 stages as the first layer, Bit-reorganization (BR) for the middle layer, a nonlinear function F the bottom layer.

The LFSR has 16 of 31-bit cells (s0, s1, …, s15) Each register takes values from {0,1,……., 231-1}. In the key filling procedure 128-bit Initial key and 128-bit initial vectors 16 bytes each other: k = k0||k1||……||k15 and IV = IV0||IV1||……||IV15. Then load into the registers of LFSR as follows: si = ki||Di||IVi ($0 \leq i \leq 15$). Here, Di is a 15-bit constant.

In the initialization, the LFSR takes a 31-bit input word u, which is gained by eliminating the rightmost bit from the 32-bit output W of the nonlinear function F, (u=W>>1). More specifically, the initialization mode works as follows:

LFSR With Initialization Mode (u)

1 begin

2 $u = 2^{15}S_{15} + 2^{17}S_{13} + 2^{21}S_{10} + 2^{20}S_4 + (1 + 2^8)S_0 \bmod (2^{31} - 1);$

3 $S_{16} = (v + u) \bmod (2^{31} - 1);$

4 *if* $S_{16} = 0$ *then*

5 *then set* $S16 = 2^{31} - 1$

6 $(S1, S2, \ldots, S15, S16) \rightarrow (S0, S1, \ldots, S14, S15);$

In the working mode, the LFSR does not receive any input, and works as follows:

LFSR With Work Mode

begin

1 $u = 2^{15}S_{15} + 2^{17}S_{13} + 2^{21}S_{10} + 2^{20}S_4 + (1 + 2^8)S_0 \bmod (2^{31} - 1);$

2 *if* $S_{16} = 0$ *then*

3 Then set $S_{16} = (2^{31} - 1)$

4 $(S1, S2, \ldots, S15, S16) \rightarrow (S0, S1, \ldots, S14, S15);$

The bit-reorganization layer fetched 128-bit from the cells of the LFSR and produce four 32-bit words, where the first three will be used by the nonlinear function F in the bottom layer and the last word will be integrated in producing the key stream. Let S_0, S_2, S_5, S_7, S_9, S_{11}, S_{14}, S_{15} be eight cells of LFSR. Then the bit-reorganization create four 32-bit words X0, X1, X2, X3 from the above cells as follows: $X0=S_{15}H \parallel S_{14}L$, $X1=S_{11}L \parallel S_9H$, $X2=S_7L \parallel S_5H$ and $X3=S_2L \parallel S_0H$ with respect at the rule that S_iH means the bits 30...15 and S_iL means the bits 15…0 of si respectively. The nonlinear function F has two 32-bit memory cells R1 and R2. Let the inputs to F be X0, X1 and X2, which arrive from the outputs of the bit-reorganization. Then function F output a 32-bit word W. The 32x32 S-box S is composed of four 8x8 mini S-boxes, i.e., S= (S0, S1, S2, S3), where S0=S2, S1=S3. The definitions of S0 and S1 can be established in the official cipher specifications. L1 and L2 are linear transformations from 32-bit words to 32-bit words.

The complete process of F is as follows:

Input: X0, X1, X1

ISSN 2348 – 8034
Impact Factor- 3.155

begin

1. W = (X0 ⊕R1) + R2;
2. W1 = R1 + X1;
3. W2 = R2 ⊕X2;
4. R1 = S (L1 (W1L || W2H));
5. R2 = S (L2 (W2L || W1H));

Where,

S is a 32 X 32 S-box,

L1, L2 are linear transformations.

For the cipher method firstly the key loading procedure expands the initial key and the initial vector into 16 of 31-bit integers as the initial state of the LFSR and then two stages are executed; initialization stage and working stage. In the first stage, a Key IV initialization is performed and the cipher is clocked without producing output. The second stage is a working stage in which every clock cycle produces a 32-bit word of output.

V. ZUC ARCHITECTURE

The aim of the work is to discover that the ZUC stream cipher can operate on a recent hardware device for efficient use on LTE networks. The hardware implementation of the ZUC stream cipher is illustrated in Fig. 1.

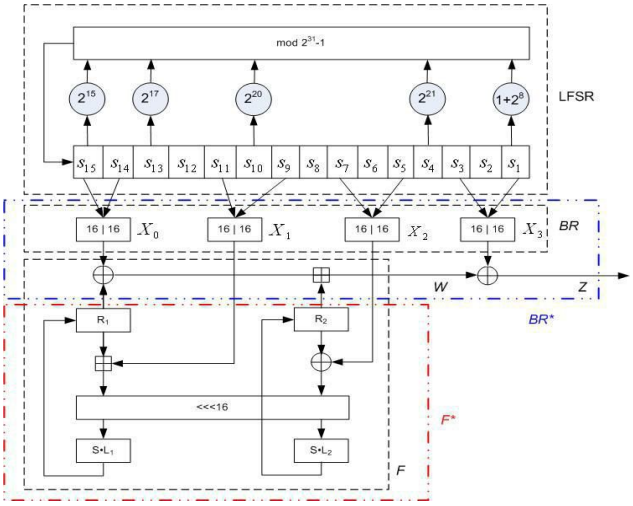

Fig 2 The structure of ZUC

The proposed system has as main I/O interfaces a 32-bit plaintext/ciphertext input and a 32-bit ciphertext/plaintext output. As a set of control logic modify, the configuration of the planned hardware system helps every stages of action. In addition it has two inputs, a 128-bit secret key, Key, and 128-bit initialization value, IV. Our system supports, the initialization stage, the working stage and the keystream producing stage. The core parts of the proposed architecture of ZUC are the Key Loading, the Linear Feedback Shift Register (LFSR), the BR (bit-reorganization) and the nonlinear function F. Lastly, a Control Unit (that is not shown in the figure) is accountable for the exact operation of the stream cipher.The Key Loading part utilize a 240-bit D constant, D = d0 ||d1 ||…||d15 (where 0 ≤ di ≤ 15 are predefined) and together with Key and IV, create 16 substrings of 31-bit according to the following rule si = ki || di || ivi (0 ≤ i ≤ 15).

The ki and ivi are considered the 16 bytes of the Key and IV correspondingly where k0 and iv0 are the mainly significant ones. Those substrings are used as initial values of the 31-bit LFSR cells S0,S1, …, S15 respectively. The substrings si are parallel loaded as the LFSR initial values through the OR gates as in Fig. 2. When the values are fetched the OR-gates are forced by zeros.

The BR compose of four simple components that perform concatenations according to the law described previously. Only wirings are used in hardware.

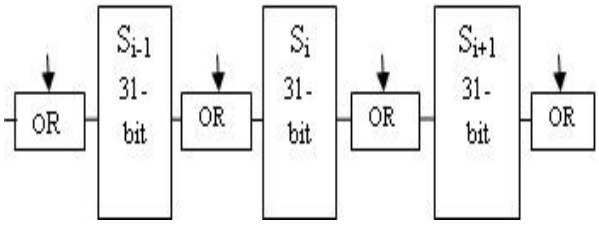

Fig 3 The 31 bit OR gates between the LFSR cells

The function F has two 32-bit registers R1 and R2, two S-boxes, two 32-bit XOR, two32-bit mod 232 adders, two linear transformations L1 and L2 and finally a left cyclical shifter of 16 positions. The transformation L1 performs the operation.

$L_1(X) = X \oplus (X <<_{32} 2) + (X <<_{32} 10) \oplus (X <<_{32} 18) \oplus (X <<_{32} 24)$

while the L_2 performs the operation

$L2(X) = X \oplus (X <<<_{32} 8) \oplus (X <<<_{32} 14) \oplus (X <<<_{32} 22) \oplus (X <<<_{32} 30)$.

So each transformation uses four 32-bit XOR-gates and four left cyclical shifters.

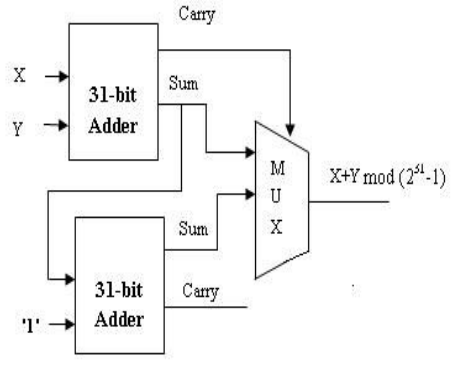

Fig. 4 The Architecture of Adder mod (2^{31}-1).

The Feedback Logic is an arithmetic logic that combines cyclical shifters and additions mod

ISSN 2348 – 8034
Impact Factor- 3.155

(2^{31}-1). The multiplexer (MUX) changed their configuration according the cipher operation situation (initialization or working stage). Also, one more adder mod (2^{31}-1) is wanted with its result used as first input of the multiplexer. In the Feedback Logic six additions mod (2^{31}-1) are used. The architecture for the two inputs, X, Y, adder mod (2^{31}-1) is depicted in Fig. 3. One adder is used in order to add the values of S0 and 2^8S0, another for the addition of 2^{20}S4 and 2^{21}S10 and another for the addition of 2^{17}S13 with 2^{15}S15. Finally, a three-input adder mod (2^{31}-1) is used to add the three preceding sums. For the three-input adder mod (2^{31}-1) two cascaded two-input address mod (2^{31}-1) are used. The circuit that evaluates the Feedback Logic is described in Fig. 4.

$(S_0+2^8S_0+2^{20}S_4+2^{21}S_{10}+2^{17}S_{13}+2^{15}S_{15})$ mod $(2^{31}-1)$

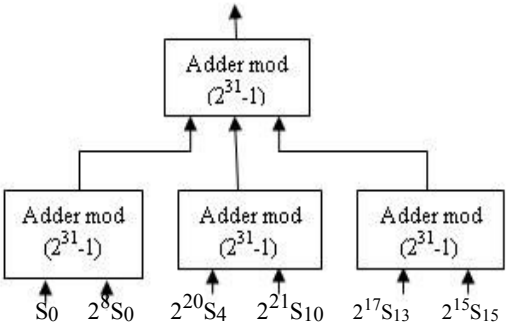

Fig. 5 The feedback logic circuit

The process of the proposed ZUC plan (see Fig. 1) starts with the first concurrent loading of the LFSR initial values. Also, the values of R1 and R2 registers are put equal to zero. Then, during the initialization stage, the LFSR receive a 31-bit word as input through the multiplexer MUX (input 1 of the multiplexer is selected). This input is produced by the addition mod (2^{31}-1) among the 31-bit output of the function F called W (the rightmost bit of the output W is detached, W>>1) and the output of the feedback logic.

During this operation the cipher is clocked without generating output. For this cause a 32-bit output register is placed at the output of the cipher that hold the formed data. In addition, during the working mode, the LFSR does not get any new input and input -2 of the multiplexer is preferred. The cipher is evaluated once, and the output W is redundant. After that, the cipher produce a 32-bit keystream Z for each clock cycle.

The keystream produced by bit-by-bit XOR between the W and X3 word that is output of BR layer. In this stage of operation the 32-bit output register latches its input to the output.

VI. CONCLUSIONS

In this system, FPGA device is used for implementation of reconfigurable ZUC hardware architecture. It uses Carry Look Ahead Adder which is the highest speed adder as compared to other. The implementation on FPGA achieved a throughput of 3.2180 Gbps. The system will be implemented for data security.

VII. ACKNOWLEDGEMENT

The authors wish to thank the anonymous reviewers. The author would like to thanks of Dr. D. Y. Patil College of Engineering, Savitribai Phule Pune University, Talegaon , Pune, Maharashtra, INDIA.

REFERENCES

1) *"A Flexible and Energy-Efficient Reconfigurable Architecture for Symmetric Cipher Processing" by Bo Wang, Leibo Liu, Institute of Microelectronics Tsinghua University Beijing, China 978-1-4799-8391-9/15 in 2015 IEEE.*

2) *"A Very High Security Cryptosystem Architecture for Video Application" by Toan Nguyen Van, Thuan Huynh Huu Faculty of Electronics and Telecommunications HCMC University of Science Ho Chi Minh City, Vietnam in 2014 IEEE.*

3) *"Evaluating the Optimized Implementations of SNOW3G and ZUC on FPGA" by Lingchen Zhang , Luning Xia , Zongbin Liu, Jiwu Jing published in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on 25-27 June 2013, Page(s): 436 – 442 ,Print ISBN: 978-1-4673-2172-3.*

4) *"Power analysis and optimization of the ZUC stream cipher for LTE-Advanced mobile terminals" by Traboulsi, S.,Pohl, N., Hausner, J., Bilgic, A. published in Circuits and Systems (LASCAS), 2012 IEEE Third Latin American Symposium on Feb. 29 2012-March 2 2012, Page(s):1 - 4 , Print ISBN: 978-1-4673-1207-3.*

5) *"Analytical evaluation of the stream cipher ZUC" by Orhanou G.,El Hajji S.,Lakbabi A.,Bentaleb Y. published in Multimedia Computing and Systems (ICMCS), 2012 International conference on date of conference: 10-12 May 2012 Page(s): 927 - 930 Print ISBN:978-1-4673-1518-0.*

6) *"An FPGA Implementation of the ZUC Stream Cipher" by Kitsos, P., Patras, Greece, Sklavos, N., Skodras, A.N. published in Digital System Design (DSD), 2011 14th Euromicro Conference on Date of Conference:Aug. 31 2011-Sept. 2 2011, Page(s): 814 – 817, Print ISBN:978-1-4577-1048-3.*

ISSN 2348 – 8034
Impact Factor- 3.155

7) *"Evaluating Optimized Implementations of Stream Cipher ZUC Algorithm on FPGA" by* Lei Wang, Jiwu Jing, Zongbin Liu, Lingchen Zhang, Wuqiong Pan *published in Springer Berlin Heidelberg 13th International Conference, ICICS 2011, Beijing, China, November 23-26, 2011.*
8) *"Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3" Document 1: 128-EEA3 and 128-EIA3 Specification; Version: 1.5, 2011.*
9) *"Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3" Document 2: ZUC Specification; Version: 1.5, 2011.*