Compact and High Speed Hardware Implementation of the Block- Cipher Clefia

V.A. Suryawanshi
Research Scholar
GHRCE Nagpur
India

G.C. Manna
Sr. General Manager
BSNL Jabalpur
India

S.S. Dorale, PhD
H.O.D Electronics Engg
GHRCE Nagpur
India

ABSTRACT
Main fundamental directions which are considered as important for practical ciphers are (1) security, (2) speed, and (3) cost for implementations. To realize these fundamental directions CLEFIA is designed. Clefia is a first block cipher employing the Diffusion Switching Mechanism (DSM) to enhance the immunity against the differential attack and the linear attack. Clefia uses lightweight components for efficient software and hardware implementations. This paper proposes compact and high speed hardware implementation for block cipher clefia-128. This hardware architecture uses minimum hardware resources and maximum frequency of 135.452 Mhz, through which we can achieve a throughputs of 17 Gbit/s

Keywords
Clefia, DSM, Encryption, FPGA and VHDL

1. INTRODUCTION
Digital data is constantly being transmitted through public open channels, In this current digital communication world, whether it is an internet network access or a through the air communication. To accomplish privacy and access management to such media, ciphering mechanisms need to be employed when sending sensitive information through these public channels.

CLEFIA [1,2] is a 128-bit block cipher supporting key lengths of 128, 192 and 256 bits which is developed by Sony Corporation in 2007. Compact hardware implementations play a key role for small embedded devices such as RFID tags and wireless sensor nodes because of their limited hardware resources. In the current digital communication world, digital data is constantly being transmitted through public open channels, whether it is an internet network access or a through the air communication, like in wireless or mobile phone networks. In order to have privacy and access management to that same media, ciphering mechanisms need to be employed when sending sensitive information through these public channels. Ciphering algorithms have been in use for a long time, but the growing processing capabilities of digital equipment and the growing bandwidth for digital communication channels impose the need for more dedicated and secure algorithms. These algorithms can be divided in two classes, asymmetric and symmetric. While the first ones are based on complex mathematical problems, thus having long processing times, the second ones are implemented using operations such as byte substitution, bit permutation and basic arithmetic operations, and can process large amounts of data in small amounts of time.

This algorithm improves the security of encryption with the use of techniques such as Diffusion Switch Mechanisms, consisting of multiple diffusion matrices in a predetermined order, to ensure immunity against differential and linear attacks [3,4,5], and the use of Whitening Keys, combining data with portions of the Key before the first round and after the last round.

In this research work FPGAs are selected as the target technology for their advantages in computation adaptability, time to market, development costs, and deployment time of dedicated solutions [6,7].

The rest of this paper is arranged like this: Section II is a brief literature review. In section III Hardware Implementation (CLEFIA-128) is presented. Section IV presents implementation results. Section V is conclusion of the paper.

2. LITERATURE REVIEW
Article of author [8] presented a paper on Very Compact Hardware Implementations of the Block cipher CLEFIA. They have implemented three types of hardware architectures and synthesized using a 0.13 µm standard cell. In the smallest implementation, the area requirements are only 2,488 GE, In article [9] compact clefia implementation on FPGAS presented by Paulo Proença and Ricardo Chaves. Throughputs above 1Gbit/s can be achieved with a resource usage as low as 86 LUTs and 3 BRAMs on a VIRTEX 5 FPGA. Implementation results suggest that a LUT reduction up to 67% can be achieved at a performance cost of 17% on a VIRTEX 4 FPGA, resulting in Throughput/Slice efficiency gains up to 2.5 times. Clefia block cipher implemented on with a 90-nm CMOS standard cell library in terms of ASIC implementation. The highest hardware efficiency (defined as throughput/gates) obtained was 400.96 Kbps/gates [10].

Pipeline implementation of the 128-bit block cipher clefia in fpga presented by Tomasz Kryjak and Marek Gorgoń The article proposes the implementation of a key scheduler, which is highly compatible with pipeline encryption. The article contains a detailed analysis of the data processing path for the 128-bit key version of the algorithm and verifies its operation on two FPGA cards in practice [11].

3. HARDWARE IMPLEMENTATION OF CLEFIA
The CLEFIA is a 128 bit block cipher symmetrical encryption algorithm with a Key size varying from 128, 192, to 256 bits. It consists of a Key Scheduling Part and a Data processing part computed in multiple rounds, allowing it to be easily implemented in platforms with limited resources [12]. The numbers of rounds of CLEFIA are 18, 22 and 26 for 128-bit, 192-bit and 256-bit keys, respectively. Generalized Feistel network GFNd,r (d-branch and r-round) is a fundamental structure for Clefia algorithm. This Generalized Feistel network GFNd,r uses two different 32-bit F-functions F0 and F1. The main goal on this research work was to provide a compact hardware CLEFIA structure, while still being able to achieve implementations with adequate throughput and performance, even on low cost devices.

3.1 Data processing part*:*

CLEFIA employs a generalized Feistel structure which is an extension of the traditional Feistel structure. Generalized Feistel structures have three or more data lines as opposed to two data lines in traditional Feistel structure.

There are many types of generalized Feistel structures depending on the connected positions of the input and the output of F-functions to the data lines. Among them, clefia choose one structure which is known as Generalized 4-branch type-2 generalized structure" defined in Zheng et al.'s paper [13] The data processing part employs a 4-branch type-2 generalized Feistel network with two parallel F-Functions F0 and F1 per round. The encryption function takes 128-bit data as a plaintext P=P0|P1|P2|P3, 32 bit whitening keys Wk1, Wk2, Wk3 and Wk4 and 32 bit 36 round keys as a input and produces 128 bit data as a cipher text C=C0|C1|C2|C3 as shown in fig 1.

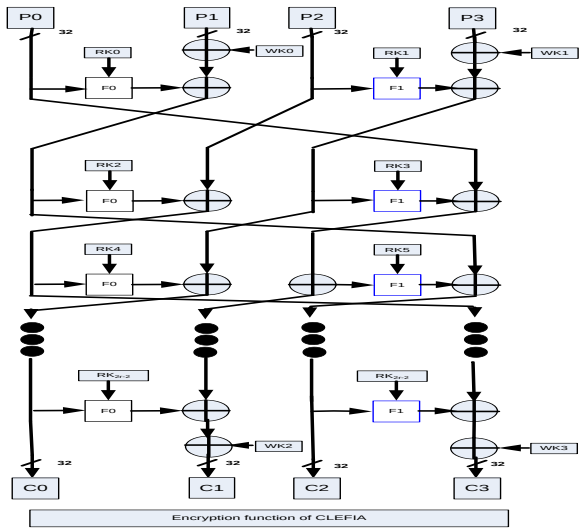


Fig 1. Encryption function Of Clefia

F-Functions F0 and F1, have different Diffusion Matrices,

providing CLEFIA with a diffusion switch mechanism. These two F-Functions are used for data randomization. These F-Functions consist of additions in GF(2^8) between the round data and the round keys, substitution boxes S0 and S1 and diffusion matrices M0 and M1 one for each function (F0 and F1) as shown in fig 2 and fig 3. Two different types of 8-bit s-boxes are used in each F-function S0 and S1. Diffusion matrices (4×4) M0 is used in F0 function and M1 is used in F1 function. These two matrices are an integral part of the diffusion mechanism present in Clefia providing with resistance to differential attacks. The diffusion matrices M0 for F0 and M1 for F1 are defined as

$$M0 = \begin{pmatrix} 01 & 02 & 04 & 06 \\ 02 & 01 & 06 & 04 \\ 04 & 06 & 01 & 02 \\ 06 & 04 & 02 & 01 \end{pmatrix} \qquad M1 = \begin{pmatrix} 01 & 08 & 02 & 0A \\ 08 & 01 & 0A & 02 \\ 02 & 0A & 01 & 08 \\ 0A & 02 & 08 & 01 \end{pmatrix}$$

These matrices and vectors multiplications are performed in GF(2^8) which is defined by primitive polynomial z^8+z^4+z^3+z^2+1.

In data processing part the 128 bit data is separated in 32 bit data as P0, P1, P2 and P3. This 32 bit data in P0 is again separated in 8 bit data which will be Xored with 8 bit of 32 bit round keys which is generated in key scheduling part on which S0 substitution box is implemented. This output is named as Y0. Next 8 bit data from P0 is Xored with 8 bit data of round key on which S1 substitution box is implemented. This output is named as Y1. Similarly Y2 and Y3 outputs are obtained by implementing S0 and S1 substitution box. The nonlinear 8-bit S-boxes are S0 and S1. This output Y0, Y1, Y2 and Y3 are multiplied with matrices m0 which is shown in fig 2. Similar operation is carried for function F1 same output Y0, Y1, Y2 and Y3 are obtained which are again multiplied with matrices m1 as shown in fig 3.

3.2 Key scheduling part

The total number of RKi depends on the key length. The Data processing part requires 36, 44 and 52 round keys for 128-bit, 192-bit and 256-bit keys, respectively. Key scheduling part generates two types of keys whitening keys and round keys respectively for the data processing part. Whitening keys are used at the beginning and the end of the data processing part. The given key is divided in 32 bit data and named as wk0, wk1, wk2 and wk3. Let K be the key and L be an intermediate key. This intermediate key L is generated from given key K. 128-bit intermediate key L is generated by applying GFN4, 12 which takes twenty-four 32-bit constant values as round keys and k=k0|k1|k2|k3 as an input. The size of each constant value is 32 bit and each value is made from 16 bit constant by applying simple bit operations repeatedly. In the round key generation process of CLEFIA, the intermediate values L are updated by a Double-Swap function in every two rounds repeatedly. The double swap function is defined as for

$$X\ (128) \rightarrow Y\ (128)$$

$$Y = X\ [7\text{-}63]|X[121\text{-}127]|X[0\text{-}6]|X[64\text{-}120]$$

where X[a-b] denotes a bit string cut from the a-th bit to the b-th bit of X. 0-th bit is the most significant bit. Double swap function is shown in fig 4

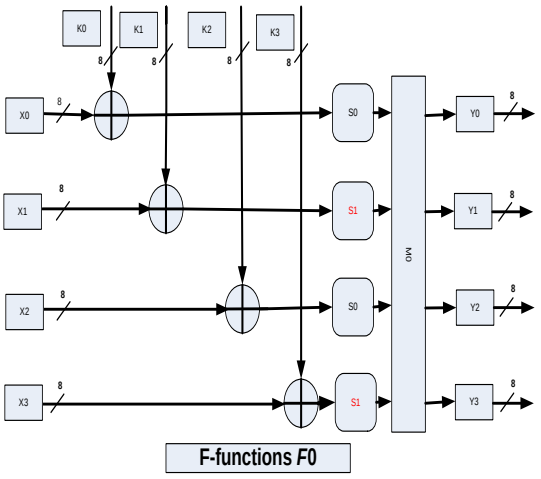
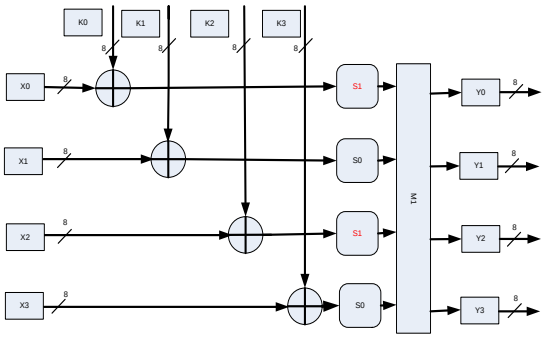


Fig 2 F-Functions F0



F-functions F1

Fig 3 F-Functions F1

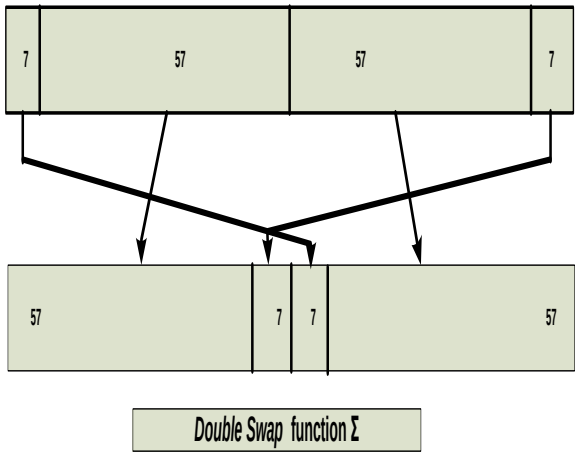


Fig 4 Double Swap Function

4. IMPLEMENTATION RESULTS

Table 1

Parameters	Used	Available	Utilization
Number of slices	3671	49152	7%
Number of slice flip-flops	2003	98304	2%
Number of 4 input Luts	6742	98304	6%
Number of Bounded IOBs	387	768	50%

Minimum period: 7.383ns (Maximum Frequency: 135.452MHz) Minimum input arrival time before clock: 2.976ns

Maximum output required time after clock: 10.455ns

Maximum combinational path delay: 4.952ns

5. CONCLUSION

In this paper hardware implementation of clefia encryption algorithm is carried out.

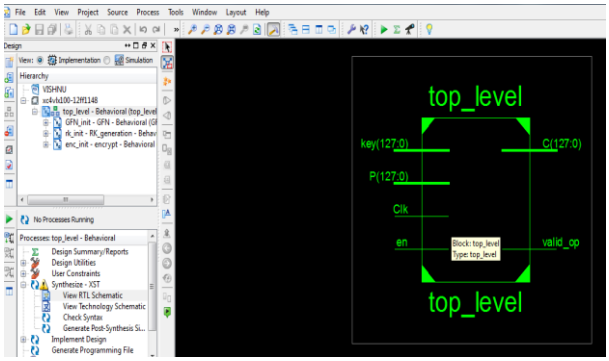


Fig 4 RTL Schematic of Hardware Implementation of CLEFIA

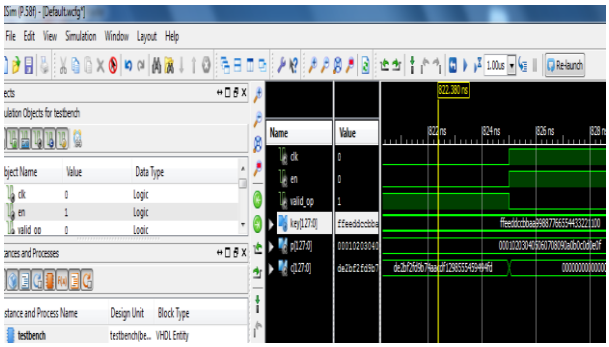


Fig 5 Output Results of Hardware Implementation of CLEFIA

6. REFERENCES

[1] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, \The 128-bit Blockcipher CLEFIA (Extended Abstract)", FSE 2007, LNCS 4593, pp. 181{195, Springer-Verlag, 2007

[2] "The 128-bit Blockcipher CLEFIA: Algorithm Speci_cation", Revision 1.0, 2007, Sony Corporation. http://www.sony.net/Products/cryptography/clefia/download/data/ clefia-spec-1.0.pdf

[3] Taizo Shirai and Kyoji Shibutani, "On Feistel Structures Using a Difusion Switching Mechanism" in Fast Software Encryption, 2006, pp. 41--56.

[4] H. and Wu, W. and Feng, D. Chen, "Differential fault analysis on CLEFIA" in Proceedings of the 9th international conference on Information and communications security, 2007, pp. 284--295.

[5] Y. and Tsujihara, E. and Shigeri, M. and Suzaki, T. and Kawabata, T. Tsunoo, "Cryptanalysis of CLEFIA using multiple impossible differentials" , 2009, pp. 1--6.

[6] Francisco Rodriguez-Henriquez, N.A. Saqib, A. Diaz-Perez, and Çetin Kaya Koç, Cryptographic Algorithms on Reconfigurable Hardware.: Springer, 2006.

[7] AJ Elbirt, W. Yip, B. Chetwind, and C. Paar, "An FPGAImplementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists" in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, 2000, pp. 13--27.

[8] Toru Akishita and Harunaga Hiwatari "Very Compact Hardware Implementations of the Block cipher CLEFIA" Sony Corporation

[9] Paulo Proença and Ricardo Chaves "COMPACT CLEFIA IMPLEMENTATION ON FPGAS" 2011 21st International Conference on Field Programmable Logic and Applications 978-0-7695-4529-5/11 IEEE DOI 10.1109/FPL.2011.101

[10] Takeshi Sugawara, Naofumi Homma, Takafumi Aoki and Akashi Satoh "High-performance ASIC Implementations of the 128-bit Block Cipher CLEFIA"

[11] Tomasz Kryjak and Marek Gorgoń "PIPELINE IMPLEMENTATION OF THE 128-BIT BLOCK CIPHER CLEFIA IN FPGA" 978-1-4244-3892-1/09/2009 IEEE

[12] T. Shirai and A. Mizumo, "A Compact and High-SpeedCipher Suitable for Limited Resources Environment" in 3rd ETSI security wrokshop presentation, Sophia-Antipolis, France, 2007.

[13] Y.Zheng, T. Matsumoto, and H. Imai, "On the construction of block ciphers provably secure and not relying on any unproved hypotheses." In Proceedings of Crypto'89 (G. Brassard, ed.), no. 435 in LNCS, pp. 461-480, Springer-Verlag, 1989.