A TIME AND AREA EFFICIENT HARDWARE IMPLEMENTATION OF THE MISTY1 BLOCK CIPHER

P. Kitsos, and O. Koufopavlou
VLSI Design Laboratory,
Electrical and Computer Engineering Department,
University of Patras, Patras, Greece.
e-mail: pkitsos@ee.upatras.gr

Abstract *- A time and area efficient hardware implementation of the 64-bit NESSIE proposal, MISTY1 block cipher, is presented in this paper. The new proposed architecture achieves high-speed and small silicon area. The VLSI implementation uses feedback logic and inner pipeline with negative edge-triggered register. So, the critical path is shorter, without increasing the latency of cipher execution. Comparing with an implementation without negative edge-triggered register, about 97 % performance improvement is achieved. The proposed implementation reaches a data throughput value equal to 561 Mbps at 79 MHz clock frequency. In addition, is area efficient because only one round of the cipher is used. The design was coded using VHDL language and for the hardware implementation FPGA device was used. A detailed analysis, in terms of performance, and covered area is shown.*

I. INTRODUCTION

With the rapid growth of the wireless standards, the subject of security in mobile communications has gained more importance. Encryption algorithms are meant to provide secure communications applications. New encryption algorithms have to operate efficiently in a variety of current and future applications, doing different encryption tasks.

Many attempts have been taken place in order to put forward new qualitative cryptography methods. The New European Schemes for Signatures, Integrity, and Encryption (NESSIE) project [1] had as object to put forward a portfolio of strong cryptographic primitives of various types. The 64-bit block cipher NESSIE winner algorithm is MISTY1 [2].

In the third NESSIE workshop [1], a paper for some NESSIE proposal algorithms was presented [3]. In this evaluation only the encryption mode of operation was implemented and not the decryption. Except of this work, for the implementation of the MISTY1 block cipher some other implementations were published [4, 5]. The proposed work in [4] is exactly the same as the MISTY1 implementation proposed in [3]. As mentioned above these implementations were not support the decryption mode of the cipher. In [5], two MISTY1 software implementations on a Digital Alpha processor were proposed. But, it is well known that the software implementations are much slower than the hardware ones.

In this paper, an architecture and VLSI implementation of the 64-bit NESSIE proposal MISTY1 block cipher is proposed. This design implements both encryption and decryption modes in the same hardware module opposed to the previous MISTY1 published implementations. This architecture can be used in applications with constrained hardware resources. It uses feedback logic and inner-pipeline with negative edge-triggered register [6, 7]. So, it makes the critical path shorter, without increasing the latency of the cipher execution. As result a significant improvement in term of performance is achieved. This architecture is suitable for feedback mode of cipher operations.

The paper is organized as follows: In section II, the MISTY1 block cipher is briefly introduced. In section III, the proposed architecture and VLSI implementation are presented and explained in details. The synthesis results and evaluation are given in section IV, and finally, section V concludes the paper

II. MISTY1 BLOCK CIPHER

The MISTY1 [2] block cipher operates with 64-bit block size plaintext and 128-bit secret key. The appropriate 64-bit ciphertext is produced after a number of n rounds, where n is a multiple of four. In [8] a number n=8 is recommended for use in real applications. Two are the main parts of the MISTY1 block cipher, the data randomizing and the key scheduling. The data randomizing part it consists of 8 identical stages (rounds) with an additional subround. In the encryption mode operation, the 64-bit plaintext is transformed into the 64-bit ciphertext by applying bitwise XOR operations and the sub-functions FOi ($1 \le i \le 8$) and FLi ($1 \le i \le 10$). In the begin the 64-bit plaintext is divided into two 32-bit strings, the left and the right. The subfunction FOi uses a 48-bit sub-key KIi and a 64-bit sub-key KOi. The subfuntion FLi uses a 32-bit sub-key KLi. The output of each round (stage) is produced by the following equations.

For the odd rounds (i = 1, 3,…, 7) :

Right string: $R_i = FL(L_{i-1}, KL_i)$ and

Left string: $L_i = FL(R_{i-1}, KL_{i+1}) \oplus FO(L_i, KO_i, KI_i)$

For the even rounds (i = 2, 4,…, 8) :

Right string: $R_i = L_{i-1}$ and

Left string: $L_i = R_{i-1} \oplus FO(L_i, KO_i, KI_i)$.

For the last round (i = 9) :

Left string: $R_9 = FI(L_8, KI_9)$ and

Right string: $L_9 = FI(R_8, KI_{10})$.

The final 64-bit ciphertext is produced from the concatenation of L9 and R9.

The decryption mode operation of MISTY1 is similar to the encryption mode. The only differences are the reverse order of the sub-keys and the replacement of the function FL by the function FL^{-1}. All the sub-keys are generated by the key scheduling part.

III. Proposed Architecture and VLSI Implementation

Usually each proposal of a new algorithm is accompanied with a software implementation in a commonly used language. But, it is well known that the software implementations are much slower than the hardware ones. In this section the efficient hardware implementation of the MISTY1 block cipher is proposed.

In Fig. 1 the proposed architecture of the MISTY1 data randomizing part is shown. The full MISTY1 block cipher execution requires nine loops of this single round. The output of each round is used as input (through the multiplexers) of the next round. The output of the left branch is used as input in the next right branch, and the output of the right branch is used as input in the next left branch. Both encryption and decryption operation are supported.

The MISTY1 single round, first, consists of two multiplexer (MUX A) in order to the appropriate value between the Plaintext / Ciphertext or the output of the previous round is selected.

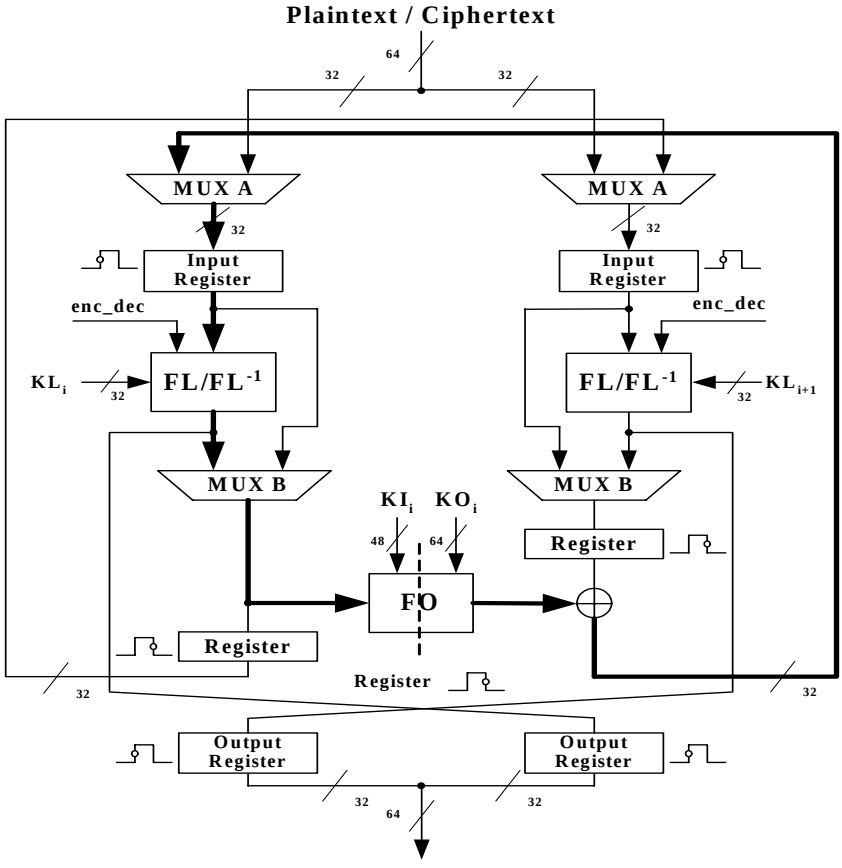

FIGURE 1

PROPOSED DATA RANDOMIZING PART ARCHITECTURE.

The input registers are necessary in order to store the input data during the operation of the FL / FL^{-1}. The FL / FL^{-1} unit is structured as the Fig. 2c shows. The enc_dec signal is used in order to determine when encryption or decryption is performed. The second layer of multiplexers (MUX B) select either the output of the FL / FL^{-1} unit or the output of the input registers, when an odd round or an even round is executed, respectively. The architectures of the FO and FI subfunctions are shown in Fig. 2a and Fig. 2b respectively.

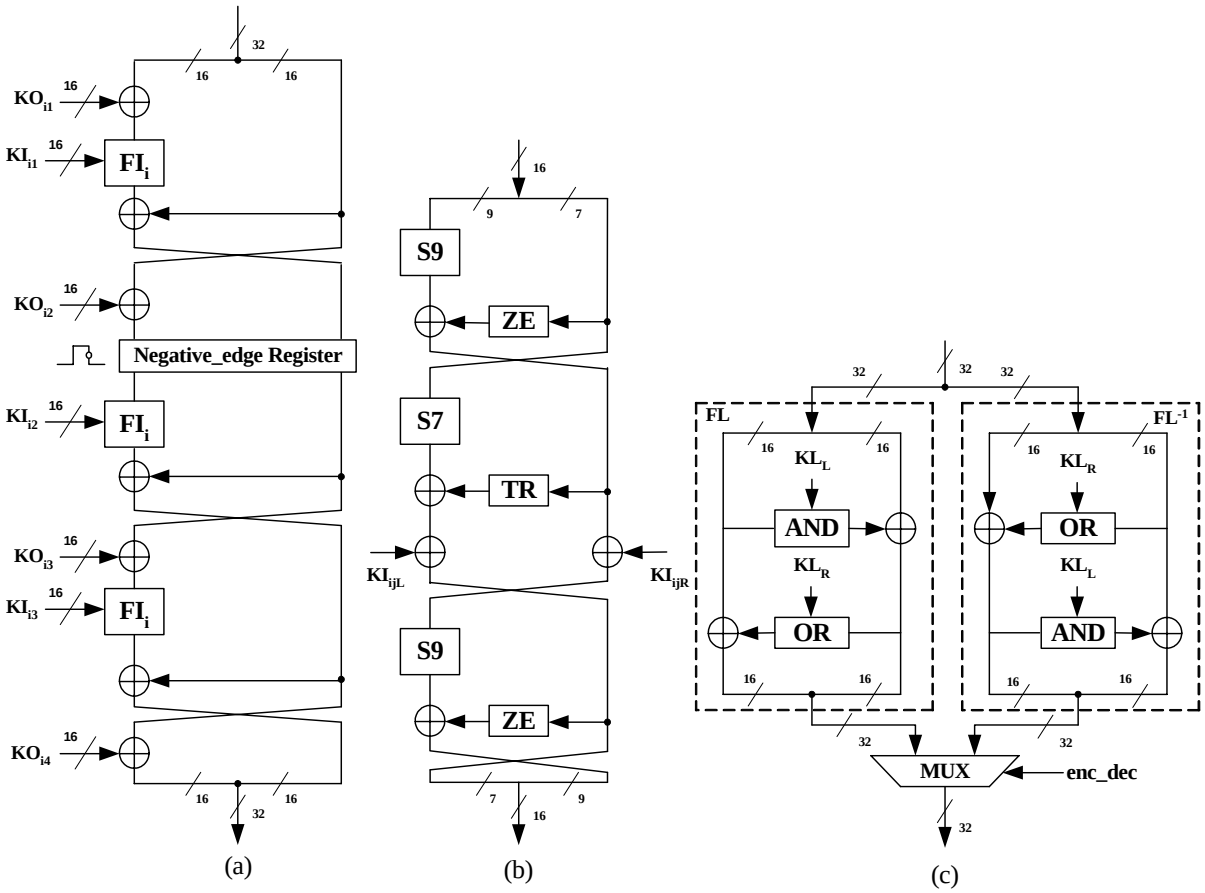

FIGURE 2
ARCHITECTURE OF THE (a) FO, (b) FI, AND (c) FL SUBFUNCTION.

In order to reduce the overall system hardware resources, for the S-boxes (S7 and S9) implementation logical expressions are used. The ZE module concatenate two zero bit ahead of the 7-bit string to 9 bits, while the TR module truncates the two most significant bits of the 9-bit string. For the FO subfunction implementation an inner pipeline with negative edge register is inserted (Fig. 1, 2a). The use of this technique (negative edge-triggered pipeline), results in a significant reduction of the round critical path delays. The negative edge pipeline register is inserted in the FO subfunction (Fig. 2a), which is roughly in the middle of the round data path (Fig. 1, heavy line).

The execution time of each round is one system clock cycle. In order to synchronize the processing data paths similar registers are inserted in the left and right branches of each round (Fig. 1). The result of this insertion is the reduction, roughly in half, of the clock period. So, the throughput is roughly doubled. A small area penalty is paid from the use of the three negative edge pipeline registers.

The usage of positive (input register) and negative edge-triggered pipeline registers (that capture data on both clock edges) demands a duty cycle of 50 %. Deviation from a 50 % duty cycle may lead to timing failures in the critical paths. The assumption of a perfect clock with 50 % duty cycle is optimistic, giving signals half the clock cycle to propagate from one register to the next. In the low level design, the duty cycle, maybe, will be not be perfect, and the actual time available for signals to propagate can be smaller. In order to avoid this problem, it must be sure to model the worst-case duty cycle of the clock accurately in synthesis and timing analysis [9].

The MISTY1 key scheduling part architecture is illustrated in Fig. 3.

The addition of extra delays is achieved in the Sub-keys Delay Unit (Fig. 3b) by using counters. In this unit, 2-input 16-bit multiplexers are used in order to the same hardware part be suitable for both encryption and decryption operations. These multiplexers are controlled by the enc_dec signal.

IV. SYNTHESIS RESULTS AND EVALUATION

The proposed architecture and implementation was captured by using VHDL, with structural description logic. The encryption and decryption operation were verified by using the test vectors provided by the NESSIE submission package [2]. The VHDL code of the design was synthesized by using FPGA devices of Xilinx [10].

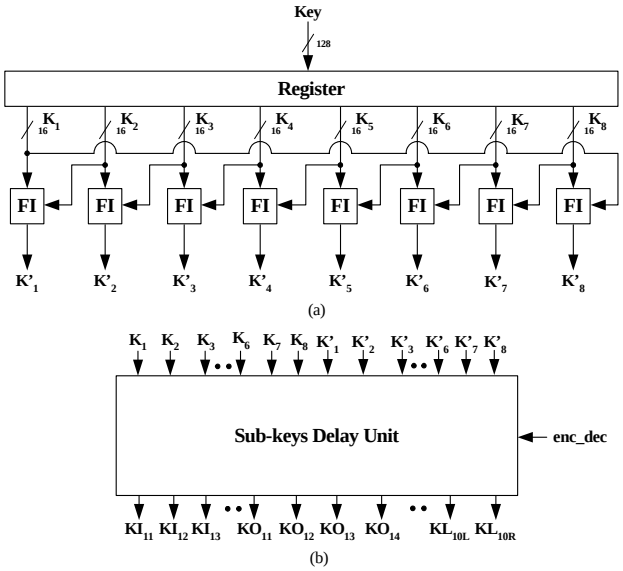

FIGURE 3
PROPOSED KEY SCHEDULING PART ARCHITECTURE.

The measurements of the performance analysis are shown in Table I. Due to no others MISTY1 implementations with feedback logic there are, comparisons with unrolled pipeline implementations are given in this table (only the faster software implementation is added [5]). However, the comparisons in terms of frequency and throughput are not fair.

TABLE I
PERFORMANCE ANALYSIS MEASUREMENTS

Architecture	FPGA Device	CLB Slices	F MHz	Throughput (Mbps)
[3]	XCV1000 BG560-6	8386	140	8960
[4]	XCV1000 BG560-6	6322	159	10176
[4]	XCVII2000 BG575-6	6322	303	19392
[5]	Software	-	500	288
Proposed	XCV400 EBG432-8	1865	79	561

The proposed (feedback logic) implementation supports both encryption and decryption operation opposed to the previous hardware implementations [3], [4]. These architectures are identical. But, the authors reported better implementation performance results in [4] than in [3]. Most probably the difference due to the better VHDL code synthesis of [4]. Finally, in [3] and [4], 208 pipeline stages are used in order to process 208 blocks simultaneously. So, these implementations are design for high-speed applications.

The hardware implementation of the proposed MISTY1 architecture was optimized with covered area constraint. For one block encryption or decryption operation nine system clock cycles are needed. In order to evaluate the influence of the inner-pipeline (negative edge-triggered register), the operation frequency was measured without the edge-triggered pipeline registers. So, 40 MHz operation frequency and 284 Mbps throughput was measured. Therefore, with the usage of these registers a 97 % performance improvement is achieved.

V. CONCLUSIONS

VLSI architecture for the design and implementation of the MISTY1 block cipher has been presented. In contrary to previous implementations, the proposed architecture support both encryption and decryption. In order to reduce the required hardware resources and the power consumption, feedback logic and inner-pipeline with negative edge-triggered register were used. The use of this register reduces the critical path of the round, without increasing the algorithmic latency. It reaches 561 Mbps throughput at 79 MHz. It is suitable for feedback mode of cipher operations and for area-constrained devices.

VI. REFERENCES

[1] "NESSIE. New European Schemes for Signatures, Integrity, and Encryption", https://www.cosic.esat.kuleuven.ac.be/nessie/

[2] Mitsuru Matsui, "Specification of MISTY1 – a 64-bit Block Cipher", *New European Scheme for Signatures, Integrity, and Encryption (NESSIE) Project*. https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions.html

[3] Francois-Xavier Standaert, G. Rouvroy, Jean-Jacques Quisquater, Jean-Didier Legat, "Efficient FPGA Implementations of Block Ciphers KHAZAD and MISTY1". *Third NESSIE Workshop*, November 6-7 2002, Munich, Germany.

[4] G. Rouvroy, Francois-Xavier Standaert Jean-Jacques Quisquater, Jean-Didier Legat, "Efficient FPGA Implementation of Block Cipher MISTY1", *10th Reconfigurable Architectures Workshop (RAW 2003)*, April 22, Nice, France.

[5] J. Nakajima and M. Matsui, "Fast Software Implementations of MISTY1 on Alpha Processors", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E82-A, No. 1, pp. 107-116, January 1999.

[6] W. Chung, Timothy Lo, and M. Sachdev, "A Comparative Analysis of Low-Power Low-Voltage Dual-Edge-Triggered Flip-Flop", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol. 10, No. 6, pp. 913-918, December 2002.

[7] A. G. M. Strollo, E. Napoli, and C. Cimino, "Analysis of Power Dissipation in Double Edge-Triggered Flip-Flops", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol. 8, No. 5, pp. 624-629, October 2000.

[8] M. Matsui, "Supporting Document of MISTY1", version 1.1. https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions.html

[9] M. Keating, and P. Bricaud, "*Reuse Methodology Manual for System-on-Chip Designs*", Kluwer Academic Publishers, 101 Philip Drive, Assinipi Park, Norwell, Massachusetts. 1999.

[10] Xilinx Inc., San Jose, Calif., "Virtex, 2.5 V Field Programmable Gate Arrays," 2003, www.xilinx.com