AN FPGA IMPLEMENTATION AND PERFORMANCE EVALUATION OF THE SEED BLOCK CIPHER

Paris Kitsos and Athanassios N. Skodras

Computer Science, Hellenic Open University, Greece
E-mails:{pkitsos, skodras}@ieee.org

ABSTRACT

An FPGA implementation of the 128-bit SEED block cipher is presented in this paper. The proposed architecture achieves high-speed with little hardware resources using feedback logic and inner pipeline with negative edge-triggered registers. In this way, the delay of the critical path is reduced, without increasing the latency of cipher execution. The proposed implementation reaches a data throughput of 369.6 Mbps at 46.2 MHz clock frequency. The design was coded using VHDL language and for the hardware implementation, the Xilinx Spartan-3A FPGA device was used.

Index Terms— Cryptographic Hardware, FPGA Implementation, ISO/IEC 18033-3 Standard, SEED Block Cipher, Embedded Security System

1. INTRODUCTION

Today's requirements for improved time performance and secure communications impose system designers the need to look for efficient hardware implementations of cryptographic algorithms. Additionally, more and more sensitive data like bank accounts, medical records, personal emails and secret keys are stored digitally and must be kept secure. For all these reasons the new encryption algorithms have to operate efficiently in a variety of current and future applications, performing different encryption tasks.

Various cryptographic algorithms such as Triple-DES [1], AES (Advanced Encryption Standard) [2], KASUMI [3] and many others have been developed to accomplish this. Another algorithm aiming at this is the SEED block cipher [4]. SEED is a cipher developed by the Korean Information Security Agency. It is used broadly throughout South Korean industry and recently has been adopted by the International Organization for Standardization (ISO/IEC 18033-3 standard [5]) for usage in a wide range of applications, both software and hardware.

In digital signal processing, the design of fast and computationally efficient algorithms has been a major focus of research activity. The objective is the design of algorithms and their respective implementation in a manner that the required computations are performed very fast.

In this paper, an efficient implementation, in terms of hardware resources / time performance of the SEED block cipher is presented. In order to reduce the required hardware resources and to improve the cipher performance, feedback logic and inner-round pipeline techniques are used. For the pipeline technique negative edge-triggered registers are used. So, the algorithmic latency time does not increase, and simultaneously the critical path is reduced roughly to half.

Recently, many designs have been proposed for the hardware implementation of the SEED block cipher [6-9]. From those, the implementations in [7-8] ASIC technologies were used. In [6] a very compact implementation is presented. Each functional module is implemented only once and is used sequentially in order to minimize the area, which however, results in small throughput. The implementation in [9] is intended for applications with high performance requirements so, the pipeline technique was used through each cipher round.

This paper is organized as follows: In section 2 the SEED block cipher is briefly described. The proposed architecture and hardware implementation are presented in detail in section 3. The synthesis results and comparison evaluation for the FPGA implementation are shown in section 4, and the paper conclusions are given in section 5.

2. SEED BLOCK CIPHER SPECIFICATIONS

The SEED block cipher [4] has been developed by KISA (Korea Information Security Agency) and is a Korean national industrial association standard. SEED is a 128-bit key symmetric block cipher that is designed to use the S-boxes and permutations that balance with the current computing technology.

The block size and the key length of SEED are 128-bit, and have the 16-round Feistel structure. A 128-bit input is divided into two 64-bit blocks (L, R), and the right 64-bit block is an input to the round function F, with a 64-bit subkey Ki generated by the key schedule. L is the most significant 64-bit of 128-bit input, and R is the least significant 64-bit. Fig. 1 summarizes the whole encryption process of the SEED block cipher.

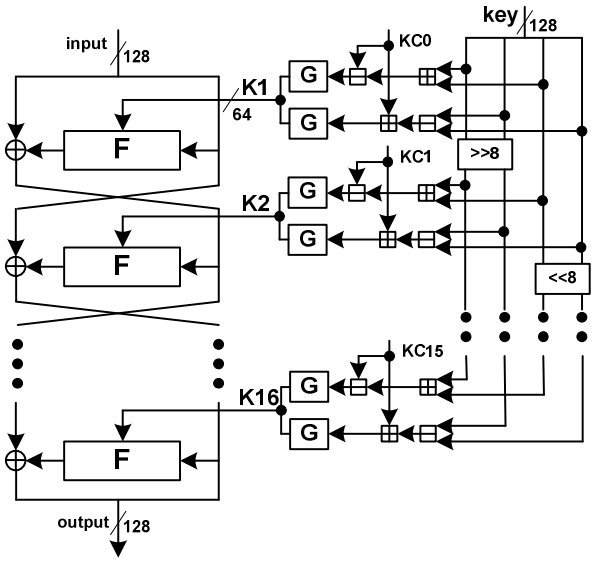

Fig. 1. The SEED block cipher architecture.

2.1. The round function F

SEED uses two S-boxes, permutations, rotations, and basic modular operations such as exclusive OR and additions to provide strong security, high speed, and simplicity in its implementation. A 64-bit input block of the round function F is divided into two 32-bit blocks (R0, R1) and wrapped in 4 phases. Firstly a mixing phase of two 32-bit subkey blocks (Ki0, Ki1) occur and then 3 layers of function G, with additions for mixing two 32-bit blocks take place. Function G has also two layers: a layer of S-boxes and a layer of XOR logic.

2.2. The key scheduling

The key schedule generates each 64-bit round's subkeys. It uses the function G, addition in modular 2^32, subtraction in modular 2^32, and (left/right) circular rotation. A 128-bit input key is divided into four 32-bit blocks (Key0, Key1, Key2, Key3).

A pseudo code for the key schedule is as follows:

Input : (Key0, Key1, Key2, Key3)

for i = 1 to 16
Ki0 = G(Key0 + Key2 - KCi)
Ki1 = G(Key1 - Key3 + KCi)

if i is odd
Key0 || Key1 = (Key0 || Key1) >> 8
else
Key2 || Key3 = (Key2 || Key3) << 8

Output : (Keyi0, Keyi1), i=1 to 16

The values of the KCi constants are given in Table I.

TABLE I
KCi CONSTANTS

i	Value	i	Value
0	0x9E3779B9	8	0x3779B99E
1	0x3C6EF373	9	0x6EF3733C
2	0x78DDE6E6	10	0xDDE6E678
3	0xF1BBCDCC	11	0xBBCDCCF1
4	0xE3779B99	12	0x779B99E3
5	0xC6EF3733	13	0xEF3733C6
6	0x8DDE6E67	14	0xDE6E678D
7	0x1BBCDCCF	15	0xBCDCCF1B

2.3. Decryption mode

The decryption mode is the reverse operation of the encryption mode. It can be implemented by using the encryption algorithm with reverse order of the round subkeys. Alternatively, the decryption mode can be implemented if the adder operations are replaced by subtractions throughout the round function F.

3. PROPOSED ARCHITECTURE AND HARDWARE IMPLEMENTATION

In Fig. 2 the proposed architecture of the SEED data randomizing part is shown. The SEED block cipher execution requires 16 loops of this single round. The execution time of each round is one system clock cycle. 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 (through the right multiplexers), and the output of the right branch is used as input in the next left branch (through the left multiplexers). The signal enc_dec determines whether encryption or decryption is performed.

The SEED round, consists of two multiplexers (MUX A and MUX B) for the selection of the appropriate value between the Plaintext / Ciphertext or the output of the previous round. The input registers are necessary in order to store the input data during the cipher operation.

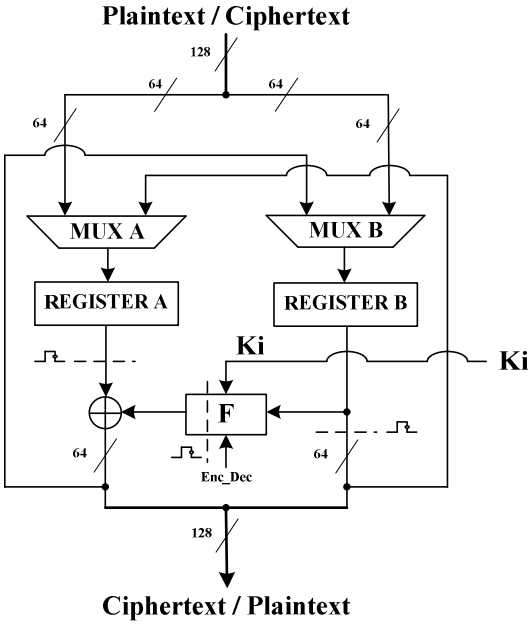

Fig. 2. Proposed architecture of the data randomizing part

Then, the round F function and a 64-bit XOR gate is used. The architectures of the F and G subfunctions are shown in Fig. 3.

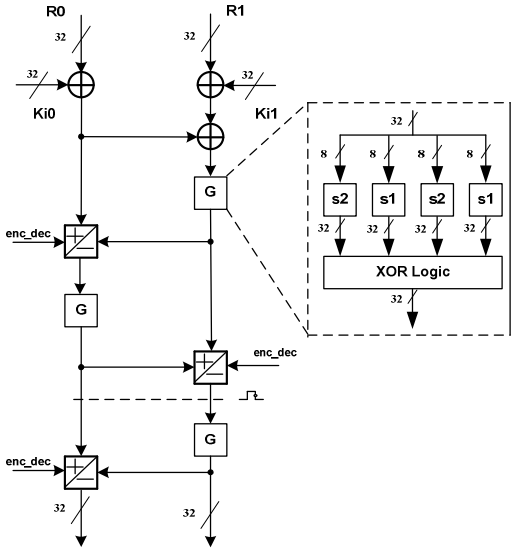

Fig. 3. Architecture of the F function and G subfunction.

The input of the round function F is divided into two 32-bit blocks (R0, R1) and executed in 4 phases. Firstly, two XOR operations of two 32-bit subkey blocks (Ki0, Ki1) with R0 and R1 blocks are performed. Then, 3 layers of G function, with additions (modulo 2^32) for mixing two 32-bit blocks take place. Function G consists of two layers: a layer of two S-boxes (the extended versions of S-boxes are used (8x32) in order to increase the efficiency of G function) and a layer of XOR logic. In order to exploit the FPGA characteristics, for the S-boxes (S1 and S2) implementation, on-chip RAM blocks are used.

For the round function F implementation an inner pipeline with negative edge register is inserted (Fig. 2, 3). The use of this technique (negative edge-triggered pipeline), results in a significant reduction of the round critical path delays without increasing the total execution latency [10]. The negative edge pipeline register is inserted in the F function (Fig. 2), which is roughly in the middle of the round data path. Also, two more 64-bit registers with negative-edge FFs are inserted (see Fig. 2) in order to synchronize the data execution through the data randomizing part. The enc_dec signal selects either addition or subtraction depending on whether encryption or decryption operation is needed.

The SEED key scheduling proposed architecture is illustrated in Fig. 4. The 128-bit cipher key is divided into four 32-bit blocks (Key0, Key1, Key2 and Key3). These blocks are processed by 16 rounds of addition/subtraction with round coefficients, 8-bit left/right rotation, and the G-function in order to generate keys for all 16-rounds. Two 32-bit negative edge-triggered pipeline registers are used before the G-function. The round keys are repeatedly generated on-the-fly by the structure in Fig. 4. Only one structure block is designed for one process of round key generation and is sequentially used. The first 64-bit round key is generated by the concatenation of two 32-bit keys, calculation of two expressions (consisting of an addition and a subtraction each one) and passing the results through the two G functions. Also, four 64-bit registers are used for the temporary storing of the key values. Each 32-bit key is concatenated with the neighboring key and rotated by 8-bits to the left or to the right in order to generate the new values of the Key0, Key1, Key2 and Key3. This process is repeated 16 times in total in order to generate the 16 round keys.

4. SYNTHESIS RESULTS AND EVALUATION

The proposed architecture was implemented in VHDL using structural description logic. The encryption and decryption operation were verified by using the test vectors provided by the ISO/IEC 18033-3 [5]. The VHDL code of the design was synthesized on a Xilinx Spartan-3A FPGA device [11-12].

The Xilinx Spartan-3A FPGA has advanced features that are useful for this application beyond traditional look-up tables (LUTs) and registers.

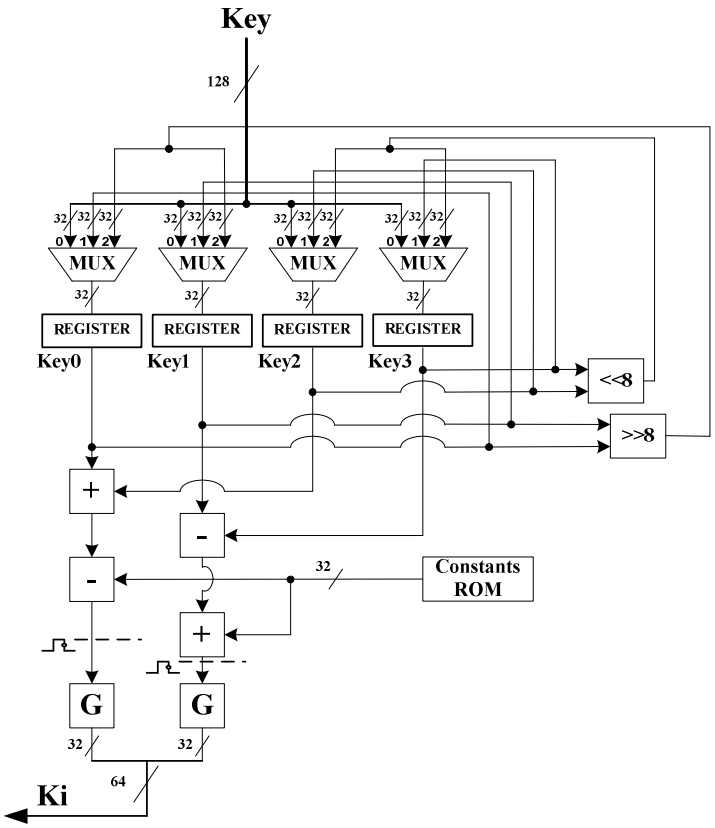

Fig. 4. Proposed architecture of SEED key scheduling.

These features are dual-ported 18-Kbit BRAMs and DSP cores (DSP48). The DSP cores allow the designer to implement timing- or resource-critical functions such as arithmetic operations on integers or Boolean expressions that would otherwise be considerably slower or resource demanding if implemented with "ordinary" logic elements.

The performance analysis results are depicted in Table II. The power consumption estimations have been produced by the XPOWER tool.

TABLE II
PERFORMANCE ANALYSIS

FPGA Device	SPARTAN3A DSP XC3SD1800 A-FG676 (-3)	
Hardware Characteristics	**Used**	**Total**
Slices	*4284*	*16640*
Flip Flops	*481*	*33280*
4 input LUTs	*8268*	*33280*
# BRAMs	*1*	*84*
# DSP48s	*8*	*84*
Clock Frequency	*46.2 MHz*	
Throughput	*369.6 Mbps*	
Total Power	*0.4 W (junction temp: 31.2 ^oC)*	

It is seen that the SEED implementation results in small hardware resource utilization. Also, for a clock frequency of 46.2 MHz, a throughput equal to 369.6 Mbps is achieved. Finally, the total power dissipation for maximum clock frequency is up to 0.4 W (the sum of dynamic and static power).

Performance comparisons between the proposed implementation and previously published works are shown in Table III.

TABLE III
COMPARISONS

Architecture	**Device**	**F (MHz)**	**Throughput (Mbps)**
SEED compact [6]	*ALTERA*	*5*	*4.4*
SEED processor1 [7]	*ASIC (Fujitsu cs66_uc-core)*	*18*	*47*
SEED processor2 [7]	*ASIC (Fujitsu cs66_uc-core)*	*40*	*35.7*
SEED (16 cycles) [8]	*ASIC (0.18 μm)*	*84.6*	*676.8*
SEED (52 cycles) [8]	*ASIC (0.18 μm)*	*170.1*	*418.6*
SEED [9]	*Virtex-V*	*50*	*6400*
Proposed SEED	*SPARTAN3A*	*46.2*	*369.6*

In [6], a compact architecture was proposed. An ALTERA FPGA was used with a clock frequency of 5 MHz and total execution of 145 clock cycles. So, the encryption rate is 4.4 Mbps. This design occupies 80% chip area of ALTERA 10KE. Also, in [7], two ASIC architectures were proposed (processor1 and processor2). The first one needed

49 clock cycles for the execution and achieved a maximum clock frequency of 18 MHz and a data throughput up to 47 Mbps. The second one needed 145 clock cycles for the execution; it achieved a maximum clock frequency of 40 MHz and a data throughput of 35.7 Mbps. The SEED processor1 occupies 24,865 ASIC gates while the SEED processor2 possesses 10,610 ASIC gates. In [8], two ASIC implementations are proposed. The first is a high speed version (with 16 clock cycles execution) and the second is a lower performance version (with 52 clock cycles execution) with reduced hardware resources. The first one achieves a throughput of 676.8 Mbps at a clock frequency of 84.6 MHz (with 36.2 KGates area) and the second one achieves a throughput of 418.6 Mbps at a clock frequency of 170.1 MHz (with 18.9 KGates area). Finally, in [9] a fully pipelined implementation is proposed. It uses sixteen cascade rounds with pipeline registers between the rounds. So, sixteen different data blocks can be processed simultaneously. It achieves a data throughput of up to 6.4 Gbps at a maximum clock frequency of 50 MHz. This implementation counts 5,314 Flip-flops and 36,678 slices LUTs.

According to Table III comparisons, the proposed SEED implementation outperforms the implementations in [6-7], and it is slower than the implementations in ASIC technology in [8]. However, it is well known that the ASIC technology is much better than the FPGA technology in terms of time performance. Finally, it is slower compared to the fully pipeline architecture. This happens because the architecture in [9] uses sixteen cascade rounds with pipeline registers between the rounds and can process sixteen different data blocks simultaneously, as opposed to the proposed architecture which uses only one round. A fair comparison in terms of area between the SEED designs is not feasible, due to the different design technologies.

5. CONCLUSIONS

A hardware architecture for the design and an FPGA implementation of the 128-bit SEED block cipher has been presented. Using feedback logic and inner-pipeline with negative edge-triggered registers, a good tradeoff between the required hardware resources and the time performance was accomplished. The use of negative-edge registers reduces the critical path of the round, without increasing the algorithmic latency. The proposed implementation uses the advanced features that the Xilinx Spartan-3A FPGA device offers, like on-chip RAM (BRAM) and DSP blocks, which result in the minimal use of traditional user logic such as flip-flops and look-up tables (LUTs). It achieves a throughput equal to 369.6 Mbps at 46.2 MHz.

6. REFERENCES

[1] Federal Information Processing Standards Publication 140-1, "*Security Requirements for Cryptographic Modules*" U. S. Department of Commerce/ NIST, Springfield, VA: NIST, 1994.

[2] Federal Information Processing Standards Publication 197: Advanced Encryption Standard (2001)

[3] KASUMI specification, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2, ETSI/SAGE, December 1999.

[4] H.J. Lee, S.J. Lee, J.H. Yoon, D.H. Cheon, J.I. Lee, "The SEED Encryption Algorithm", *RFC 4269 - Internet Engineering Task Force*, 2005.

[5] International Organization for Standardization, "*ISO/IEC 18033-3: Information technology – Security techniques – Encryption algorithms – Part 3: Block Ciphers*", 2010.

[6] Young-Ho Seo, Jong-Hyeon Kim, Dong-Wook Kim, "Hardware implementation of 128-bit symmetric cipher SEED", *Proceedings of the Second IEEE Asia Pacific Conference on ASICs, AP-ASIC 2000*, pp. 183 – 186, 2000.

[7] Hong-Mook Choi, Hwa-Hyun Cho, Sang-Kil Lee, Myung-Ryul Choi, "High performance SEED processors", *IEEE Workshop on Signal Processing Systems*, SIPS 2003, pp. 269 – 274, 2003.

[8] T. Sugawara, N. Homma, T. Aoki and A. Satoh, "ASIC performance comparison for the ISO standard block ciphers", *The 2nd Joint Workshop on Information Security*, pp. 485-498, 2007.

[9] Jaeyoung Yi, Karam Park, Joonseok Park and Won W. Ro, "Fully Pipelined Hardware Implementation of 128-Bit SEED Block Cipher Algorithm", *Reconfigurable Computing: Architectures, Tools and Applications*, Lecture Notes in Computer Science, Vol. 5453, pp. 181-192, 2009.

[10] 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.

[11] XILINX INC. 2010. UG331: Spartan-3 Generation FPGA User Guide. http://www.xilinx.com/support/documentation/user_guides/ug331.pdf.

[12] XILINX INC. 2008. UG431: XtremeDSP DSP48A for Spartan-3A DSP FPGAs. http://www.xilinx.com/support/documentation/user_guides/ug431.pdf.