FPGA Implementation of the SMS4 Block Cipher in the Chinese WAPI Standard

Xianwei Gao Erhong Lu Liqin Xian Hanlin Chen
Beijing Electronic Science and Technology Institute, Beijing 100070, P.R. China
Gaoxianwei@besti.edu.cn

Abstract

SMS4 is a 32-round block cipher with a 128-bit block size and a 128-bit user key. This paper presents rolling and unrolling field programmable gate array implementation of the SMS4 algorithm, and both the encryption and the decryption algorithms of SMS4 have been implemented on the same FPGA. The rolling design of SMS4 for area requires 1552 ALMs，the maximum operating clock is 139MHz and the corresponding data throughput is about 539 Mbit/s. The unrolling design of SMS4 for speed requires 8373 ALMs, the maximum operating clock is 162 MHz and the corresponding data throughput is about 20736 Mbit/s. Our SMS4 implementation has a good balance between high performance and low complexity in area as a result of taking advantage of certain features present in Straitx II FPGA and some design strategies.

1. Introduction

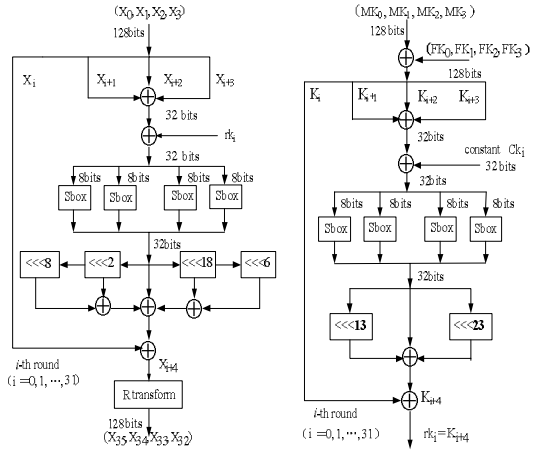
SMS4 algorithm is used in the Chinese National Standard for Wireless LAN WAPI [1]. Although WAPI was rejected In March 2006 by ISO (International Organization for Standardization), however, WAPI continues to be used in the Chinese WLAN industry. Many international corporations, such as SONY, support WAPI in relevant products.

SMS4 is a block cipher that produces a 128 bit output from a 128 bit input under the control of a 128 bit key [2]. In this paper both rolling and unrolling Field Programmable Gate Array (FPGA) implementation of the SMS4 algorithm is presented, and some advanced features of the ALMs in the Straitx II devices were also discussed.

The rest of this paper is organized as follows. In Section 2 the SMS4 algorithm is briefly described. The actual implementation of the SMS4 algorithm is presented in Section 3. The results of the FPGA implementation and evaluation are shown in Section 4, and the paper conclusions are given in Section 5.

2. Description of SMS4 Algorithm

SMS4 cryptographic algorithm is a block cipher [3], [4]. The encryption algorithm and key expansion algorithm adopt nonlinear 32 round iteration structure. The structure of the decryption algorithm is the same as the structure of the encryption algorithm. Their only difference is the order of round keys, i.e. the order of the round keys in the decryption process is inversed to that in the encryption process. Fig.1 illustrates the encryption flow and key expansion flow of SMS4.



a Encryption flow b Key expansion flow

Fig. 1. SMS4 algorithm

3. FPGA Implementation of the SMS4

FPGA offers a hardware implementation choice that is much more flexible than Application Specific Integrated Circuit (ASIC), FPGA device consist of arrays of configurable logic blocks that implement logical functions of gates and are easily reconfigurable. In contrast, ASIC provides only the functionality needed for a specific task. Therefore, we have chosen FPGA as the target technology for implementing the SMS4 cryptographic algorithm. VHDL was chosen as a language used to describe SMS4 implementation. We

978-0-7695-3288-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ICESS.Symposia.2008.76

use RTL-level VHDL coding style to describe the entire design. Probably, the best way to write the most efficient code is to use vendor supported libraries. However, this way of coding would make our design specific for a particular device family, and when multiple, fast, small RAMs(<4kb) are distributed, LUT-Based RAMS offer an ideal solution. Therefore, we have described the entire system circuit in pure VHDL'93 language, and no block RAMs are used. The functional VHDL simulation of the design is carried out using the Altera Quartus II 7.2 to verify the correct operation of the cryptographic algorithm.

As target device we have chosen a family of Straitx II devices, which are the industry's first FPGAs with the ability to decrypt a configuration bitstream using the Advanced Encryption Standard (AES) algorithm. When the design security feature is enabled, a Stratix II device must be configured with a configuration file that was encrypted using the same 128-bit security key.

The basic building block of logic in the Stratix II architecture, the adaptive logic module (ALM) shown in figure 2, provides advanced features with efficient logic utilization.

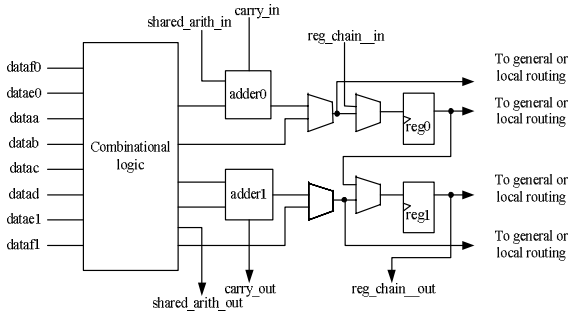
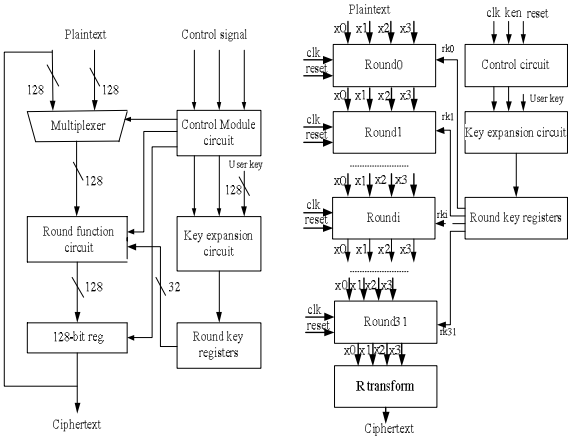


Fig. 2. Block Diagram of the Stratix II ALM

Each ALM contains a variety of look-up table (LUT)-based resources that can be divided between two adaptive LUTs (ALUTs). With up to eight inputs to the two ALUTs, one ALM can implement various combinations of two functions. This adaptability allows the ALM to be completely backward-compatible with four-input LUT architectures. One ALM can also implement any function of up to six inputs and certain seven-input functions. In addition to the adaptive LUT-based resources, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Through these dedicated resources, the ALM can efficiently implement various arithmetic functions and shift registers. The 90-nm-based Stratix II family provides an ideal design platform to efficiently implement high-performance cryptosystems.

3.1 Rolling architecture implementation

The rolling architecture shown in Fig. 3(a), which uses a feedback structure, and the data are iteratively transformed by the round functions. The control module circuit provides the necessary signals to control the flow of operations in the systems. The major component in the control circuit is a finite state machine, which was coded as a Moore machine. The key expansion circuit performs the key expansion function and the round key registers stores the round keys. For each new cipher key, the round keys are pre-calculated to allow rapid encryption of subsequent data blocks for the same cipher key, no further key expansion has to be done. Because decryption uses the encryption round keys in the reverse order, the key expansion function must be calculated only once. Hence, the stored round keys are used for both encryption and decryption. Rolling architecture approach occupies smaller area, but achieves relatively low throughput.



a Rolling architecture b Unrolling architecture

Fig. 3. Rolling architecture

3.2 Unrolling architecture implementation

Unrolling architecture is illustrated in Fig. 3(b). It uses 32 level pipeline techniques in order to maximize speed and throughput. There are 32 pipeline registers in the architecture. The registers in the first round are used to register input data. Other registers temporally store the output of the previous round. A new 128 bit data block can be loaded every clock cycle. In the 32nd clock cycle, processing of the first 128 bit data block is finished. After the 32nd clock cycle, every clock cycle processing of a new 128 bit data block can be finished. The architecture has the possibility to process 32 data blocks simultaneously in one clock cycle. So this

architecture can maximize throughput. The round keys are also generated in forward and registered.

4. The Synthesis Results and Evaluation

The synthesis results of the proposed SMS4 implementations are shown in Table 1. The rolling design for area of SMS4 require 1552 ALMs, including 998 ALUTs total with a distribution of 256 7-LUT, 474 6-LUT, 39 5-LUT, 126 4-LUT, and 103 others. The maximum operating clock is 139MHz and the corresponding data throughput is about 539 Mbit/s. The unrolling design of SMS4 for speed requires 8373 ALMs, including 7769 ALUTs total with a distribution of 7208 6-LUT, 15 5-LUT, 332 4-LUT and 214 others. The maximum operating clock is 162 MHz and the corresponding data throughput is about 20726 Mbit/s.

Table 1. Comparison of different architectures and published work

Architecture	ALMs/Device	Freq.(M)	Throughout(M bit/s)
This work (Rolling)	1552(ALMs) EP2S15F484C3	139	539
Work in [6] (Rolling)	3406(LE) EP2C35F672C6	88	342
This work (Unrolling)	8373(ALMs)/ EP2S30F484C3	162	20736
Work in [7] (Unrolling)	(?)/Virtex2	97	12400

Direct comparison among various FPGA implementations of the SMS4 algorithms is difficult, since FPGA target devices are usually different. The Stratix II family's innovative ALM-based logic structure delivers more logic capacity and faster performance in a smaller physical area. By enabling adjacent look-up tables to share logic and inputs, ALMs reduce the logic resources required for any given function and the number of logic levels needed in a given critical path. In addition, two independent functions can be packed into a single ALM, further reducing logic resource requirements. Composed of combinational, arithmetic, and register logic, an ALM is more powerful and efficient than logic structures used in previous FPGA architectures. The Quartus II software automatically utilizes the full potential of the Stratix II ALM to fit two LUTs with the same or different sizes in one ALM.

The proposed implementation outperforms previous published SMS4 implementation in terms of clock frequency and throughput [5, 6].

5. Conclusion

A high performance rolling and unrolling implementation of SMS4 is proposed in this paper. In the design, both encryption and decryption are implemented in the same module, and the round keys are generated in forward and registered. The architecture turns out to have a good balance between high performance and low complexity in area as a result of taking advantage of certain features present in Straitx II FPGA and some design strategies. Compared with previous FPGA implementations, the results show that the proposed designs outperform previous published SMS4 implementation in terms of clock frequency and throughput.

6. References

[1] Jiqiang Lu.: Attacking Reduced-Round Versions of the SMS4 Block Cipher in the Chinese WAPI Standard. In: 9th International Conference on Information and Communications Security, LNCS, vol. 4861, pp. 306--318, ZhengZhou China (2007)

[2] Office of State Commercial Cryptography Administration, P.R. China, Functionality and Interface Specification of Cryptographic Support Platform for Trusted Computing (in Chinese). http://www.oscca.gov.cn (2007)

[3] Office of State Commercial Cryptography Administration, P.R. China, the SMS4 Block Cipher (in Chinese). http://www.oscca.gov.cn (2006)

[4] Zheng Xiu-Lin, Jin Li-Na: Research of SMS4's Implementation in DSP. Journal of Beijing Electronic Science and Technology Institute, Dec.2006, vol.14, pp.34-37 (2006)

[5] Zhang Lei, Wu Wen-Ling: Differential Fault Analysis on SMS4. In: Chinese Journal of Computers, Sept 2006 vol.29, pp. 1596-1531(2006)

[6] Zhang Yuan-Yang: Area-Efficient IP Core Design Of Block Cipher SMS4, In: Electrical Technology Application, Jan. 2007, vol. 23, pp. 127-29(2007)

[7] Li Da-wei, Zhao Xu-xin, Wu Meng: Pipelined High-Speed Implementation of SMS4. In: Chinese Journal Of Electron Devices, Apr.2007, vol.30, pp.590-92(2007)