# HIGH-PERFORMANCE BLOCK CIPHER USING FLEXIBLE ARCHITECTURE

**A.SINDHU,M.E.,**

# Appusanthi1997@gmail.com

## ABSTRACT

Lightweight cryptography has been an important area of cryptographic research in recent years. In this Project, a high-throughput and flexible hardware implementations of the SPECK light weight block cipher is presented. In the SPECK block cipher, to reduce critical path delay, a tree structure for implementation of Sklansky adder which is an efficient parallel prefix adder operation is used. Simon cipher supports the variable key sizes (128, 144,192, and 256) bits and block sizes (64, 96, and 128) bits is also implemented by using the parallel prefix adder structure reduce the critical path delay.

## Key Words: cryptographic, Simon, Speck, Delay

**INTRODUCTION**

Lightweight cryptography has been an important area of cryptographic research in recent years. Different lightweight block ciphers with various design strategies have been presented.1–3 the advanced encryption standard (AES) is one of the most important and applicable block ciphers in hardware and software.4 for low-resource devices, the AES is a too expensive block cipher. Therefore, many lightweight block ciphers have been proposed to reduce the costs of hardware consumption than that of AES. Lightweight block cipher algorithms play an important role in the security for resource-constrained devices, such as radio frequency identification (RFID) tags, smart cards, and wireless sensor network (WSN) nodes. SIMON are lightweight block cipher families with 32 to 128 bits block size and 64 to 256 bits key

length, which is suitable for lightweight hardware implementation such as embedded CPUs that are used in the low-area cryptographic application systems. The first method is based on high-level synthesis to give more flexibility and to achieve suitable throughput. The second method is based on a bit-serialized structure to achieve the minimum area and cost in the design. In this method, the structure only used 34 slices on Spartan-3 FPGA. This implementation of SIMON consumed 36 slices on a Spartan- 3FPGA and 13 slices on a Spartan-6 FPGA. The focus of this paper is the design and implementation of efficient VLSI structures for the SIMON block ciphers. The throughput, execution time, and flexibility of these block ciphers are important factors for hardware implementations. Therefore, the proposed structures are implemented based on efficient components for high-throughput and versatile applications.

## PROPOSEDSYSTEM

In this subsection, we describe the proposed architectures for implementing the SPECK block cipher. In this cipher the modular adder (mod 2n) is the main block which has the direct impact on critical path delay (CPD) and hardware consumption. In the following, the structure of the modular adder based on a parallel prefix adder is presented. The proposed structure of the SPECK blocks cipher. In this design, during the first clock cycle, the Start signal is set to “1” and the 2n- bit plaintext P[2n − 1 ∶ 0] is applied to the structure and stored in two registers Reg1 and

stored in two registers Reg1 and Reg2. Then, in the second clock cycle, the *Start* signal is set to “0” for computing the following rounds.



The intermediate results are stored in the same registers. Computations in each round consist of two shift operations, two bit-wise XOR operations and one modular adder mod 2*n*. In the last step, the value of the final round key *kr*−1 is added to the outputof the modular adder for generating the 2*n*-bit cipher text. The key scheduling process in the SPECK cipher for three cases *m* = 2, 3, and 4 is shown in Figure 5A-C, respectively. The master key is divided into key words *k*0 to *km*−1. In the first clock cycle, the *Start* signal is set to “1” and the words *k*0 to *km*−1 are loaded into the registers Reg1 to Reg4, respectively. During the second clock cycle, the *Start* signal is set to “0.” In this mode, Reg1 to Reg3 are configured as shift registers and other key words *k*4 to *kr*−1are generated.

The intermediate results are stored in the same registers. Computations in each round consist of two shift operations, two bit-wise XOR operations and one modular adder mod 2*n*. In the last step, the value of the final round key *kr*−1 is added to the outputof the modular adder for generating the 2*n*-bit cipher text. The key scheduling process in the SPECK cipher for three cases *m* = 2, 3, and 4 is shown in Figure 5A-C, respectively. The master key is divided into key words *k*0 to *km*−1. In the first clock cycle, the *Start* signal is set to “1” and the words

*k*0 to *km*−1 are loaded into the registers Reg1 to Reg4, respectively. During the second clock cycle, the *Start* signal is set to “0.” In this mode, Reg1 to Reg3 are configured as shift registers and other key words *k*4 to *kr*−1are generated.

The critical path delay in these structures is reduced compared with the CRA. The depth of the adder structures is reduced on the basis of generating the carry signals in parallel form. This property is the main advantage of these adders. The structural details of the existing k- bit parallel prefix adders. The Sklansky adder has suitable hardware and time complexities compared with the others. Therefore, here, we use this adder for implementing the modular adder in the SPECK block cipher.

## RESULT



The results of the proposed implementations and other related works on speck cipher are that indicates the critical path

relay of the existing work and the proposed work of the Simon and speck block cipher using flexible architecture. This is light weight block cipher to reduce the critical path delay to perform the encryption and decryption operation. It is most useful method.

## CONCLUSION

The proposed structures are implemented on the proposed structures for SIMON; the XOR operations are implemented on the basis of a tree structure for reducing critical path delay. In addition, we proposed flexible structures that can perform various configurations of the SIMON ciphers supporting variable key and block sizes. Therefore, the flexible architectures provide versatile implementations with adaptive security level and the ability of encryption of longer messages based on variable key size and variable block size.

The implementation results show that the proposed structures have better performance in terms of critical path delay compared with other existing works. The result analysis are compared and verified by using Xilinx ISE 14.2 software tool. The future work is implemented by using SPECK block cipher with add one adder to reduce the delay and power at the output side.