RESEARCH PAPER
Engineering
Volume : 5 | Issue : 2 | Feb 2015 | ISSN - 2249-555X

FPGA Implementation of Scalable Encryption Algorithm Using Veriloghdl With Xilinx Spartan-3

KEYWORDS
FPGA – Field Programmable Gate Array, Computer security, DES - Data Encryption Standard, Verilog HDL.

Mr.K.ASHOKKUMAR	Mr.G.SENTHILKUMAR	Mrs.S. POONGUZHALI
Asst.Prof Dept Of Electronics SNR Sons College Coimbatore - 06	HEAD IN ELECTRONICS, SNR SONS COLLEGE,COIMBATORE.	ASSOCIATE PROFESSORDEPT OF ELECTRONICSSNR SONS COLLEGE, COIMBATORE.

ABSTRACT *Initially SEA is designed for software implementations in controllers, smart cards, or processors. In this Paper we proposed a system that investigates its performances in recent field-programmable gate array (FPGA) devices. The present symmetric encryption algorithms result from a tradeoff between implementation cost and resulting performances. The proposed system is applicable where there are limited processing resources with high throughput requirements. For this purpose, we propose a SEA loop Architecture with Behavior model (VerlogHDL) coding. So the number of logic gates required is very less when compared with Gate level model. Because of less number of logic gates, time taken to execute the loop architecture is less. So we are achieving a faster execution time (Frequency in MHZ). The proposed design is parametric in the key and word size, provably secure against linear or differential cryptanalysis. Beyond its low cost performances, a significant advantage of the proposed architecture is its full flexibility for any parameter of the scalable encryption algorithm, taking advantage of generic VerlogHDL coding with Xilinx Spartan 3 - 3s400ft256-5*

1 INTRODUCTION

SEA is a parametric block cipher for resource constrained systems (*e.g.* sensor networks, RFIDs) that has been introduced in [1]. It was initially designed as a low-cost encryption/ authentication routine (*i.e.* with small code size and memory) targeted for processors with a limited instruction set (*i.e.* AND, OR, XOR gates, word rotation and modular addition). Additionally and contrary to most recent block ciphers (*e.g.* the DES [2] and AES Rijndael [3], [4]), the algorithm takes the plaintext, key *and* the bus sizes as parameters and therefore can be straightforwardly adapted to various implementation contexts and/or security requirements. Compared to older solutions for low cost encryption like TEA (Tiny Encryption Algorithm) [5] or Yuval's proposal [6], SEA also benefits from a stronger security analysis, derived from recent advances in block cipher design/cryptanalysis. In practice, SEA has been proven to be an efficient solution for embedded software applications using microcontrollers, but its hardware performances have not yet been investigated. Consequently, and as a first step towards hardware performance analysis, this letter explores the features of a low cost FPGA encryption/decryption core for SEA. In addition to the performance evaluation, we show that the algorithm's scalability can be turned into a *fully generic* VerilogHDL design, using Xilinx Spartan 3 XC3S400 Device. so that any text, key *and* bus size can be straightforwardly re-implemented without any modification of the hardware description language, with standard synthesis and implementation tools. In the rest of the letter, we first provide a brief description of the algorithm specifications. Then we describe the details of our generic loop architecture and its implementation results. Finally, we discuss some illustrative comparisons of the hardware performances of SEA, the AES Rijndael and ICEBERG (a cipher purposed for efficient FPGA implementations) with respect to their design approach (*e.g.* flexible *vs.* platform/context-oriented).

2. RELATED WORK

All Scalable encryption algorithm (SEA) is a parametric block cipher for resource constrained systems (e.g., sensor networks,RFIDs) that has been introduced in [1]. SEA **n** and **b** operates on various text, key, and word sizes. It is based on a Feistel

structure with a variable number of rounds, and is defined with respect to the following parameters:

- n plaintext size, key size;
- b processor (or word) size;s
- nb: number of words per Feistel branch;
- nr number of block cipher rounds.

Let x be a n=2-bit vector. We consider the following two representations.

- **Bit representation**: xb = x((n=2) 1) _ x(2) x(1) x(0).- - - - **(1)**

- **Word representation**: xW = xn 1 xn 2 _ x2 x1 x0.- - - - **(2)**

The number of rounds nr is an optional input that can be automatically derived from n and b according to the guidelines given in [2]. A **Complete cipher** of the paper [1] is presented in the below algorithm. The cipher iterates an odd number nr of rounds. The following pseudo-C code encrypts a plain text P under a key K and produces a cipher text C. P, C and K have a parametric bit size n. The operations within the cipher are performed considering parametric b-bit words.

```
C=SEAn;b(P;K)

{

%initialization:

L0&R0 = P;
KL0&KR0 = K;

%keyscheduling:

for i in 1 to nr / 2

[KLi;KRi]= FK(KLi¡1;KRi¡1;C(i));

switch KL nr/2, KR nr/2

for i in nr/2 to nr - 1

[KLi;KRi]= FK(KLi¡1;KRi¡1;C(r ¡ i));

%encryption:

for i in 1 to nr/2

[Li;Ri]= FE(Li¡1;Ri¡1;KRi¡1);

for i in nr/2+1 to nr

[Li;Ri]= FE(Li¡1;Ri¡1;KLi¡1);

%Final:

C = Rnr&Lnr ;

switch KLnr¡1, KRnr¡1;

}
```

In this paper , we consequently consider a general context where we have very limited processing resources (*e.g.* a small Processor) and throughput requirements. It yields design criteria such as: low memory requirements, small code size, limited

instruction set. In addition, they proposed the flexibility as another unusual design principle. In opposition, SEA **n and b** allows to obtain a small encryption routine targeted to any given processor, the security of the cipher being adapted in function of its key size. Both of encryption and decryption result in an improved efficiency and are particularly relevant in contexts where the same constrained device has to perform encryption and decryption operations (*e.g.* authentication). In the paper [3] they discussed the implementation of AES and concluded that AES minimizes mean power consumption .The design of AES hardware implementation used flexible methodology which put forth a lot of possible optimization ideas. All ideas were evaluated regarding their impact on the silicon size and on the power efficiency. The evaluation is based on synthesis results and circuit-level simulations. These in-depth analyses ensure that our circuit achieves the ambitious requirements for passively powered devices A closely related work is also presented in existing design, which studies the AES implementation on Xilinx Spartan 3 family using FPGA's and also they have shown that FPGAs can be used very efficiently for high-speed implementations of cryptographic algorithms and also it can be efficiently implemented on FPGAs for applications with various requirements.

3. FPGA IMPLEMENTATION

Since we are emphasizing on the hardware implementation of the Hummingbird algorithm, so the FPGA(Field Programmable Gate Arrays) is the hardware platform selected depending on the application needs and constraints. FPGA configuration is specified using a hardware description language (HDL). Verilog is the hardware description language used for designing as well as simulation purposes. FPGAs comprises of logic blocks(flip flops, gates, memory elements) that are used to implement any logic functionality. We have described the FPGA implementation of Scalable encryption algorithm using Spartan 3 XC3S400 of Xilinx(FPGAs) for the hardware implementation of the cryptosystem. The implemented design consumes low power of 262.57 mW for 2.5 V with the operating speed or frequency of 152.905 MHz 4.

4. IMPLEMENTATION RESULTS AND COMPARISONS

The hardware design of Hummingbird on FPGA is presented using Hardware description language as Verilog via Xilinx Plan Ahead simultaneously. Virtual model of hardware is verified and simulated via. Model-Sim simulator and synthesized using Xilinx ISE suite. The design layout is presented by integrating all the components ie. Initialization module,cipher, S-box, LFSR, Encryption/Decryption and top module of the algorithm.After simulation and synthesis, the next step is place and route which provides the hardware design for the proposed Hummingbird Cryptographic Algorithm. All experimental results were extracted after place and route with the Xilinx ISE 6.0 Design Suite and the target device is 3s400ft256-5

5. HARDWARE PLATFORM DESIGN IMPLEMENTATION

The Xilinx Plan Ahead tool is embedded within the Xilinx ISE suite; it provides the post synthesis analysis and generates the hardware platform . After performing the syntax checking and RTL or technology design from the Xilinx ISE suite, the Xilinx plan ahead fetch the code directly. The hardware design of each of Hummingbird Encryption technique is implemented below in the figures: 3, 4.After the functional simulation, the Xilinx synthesis technology synthesizes VHDL or Verilog code to create Xilinx specific .ngr and .ngc files.Register-transfer level(RTL) is a simplified delegation of the pre-optimized design optimized at the register level in terms of adders , multipliers, counters, AND/ OR Gates. The RTL schematic represents the intermediate block which are monitored for the speed optimization

5.1 TECHNOLOGY SCHEMATIC

This is the schematically model of logic elements optimized to the Xilinx target device or "technology" in terms of LUT's, I/O buffers, carry logic and other technology specific components. These .ngc files contains logical data along with the constraints, fig 1 and Fig 2 is the RTL(Register Transfer Logic) diagram

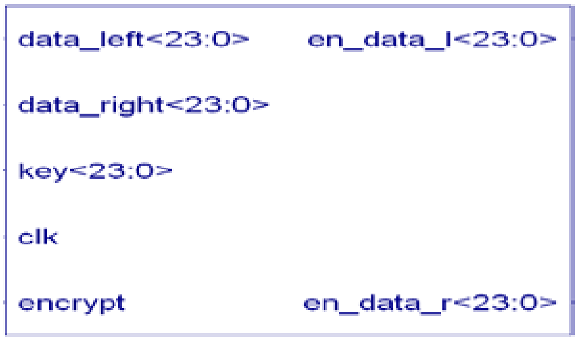

Fig 1: Top level view

Volume : 5 | Issue : 2 | Feb 2015 | ISSN - 2249-555X

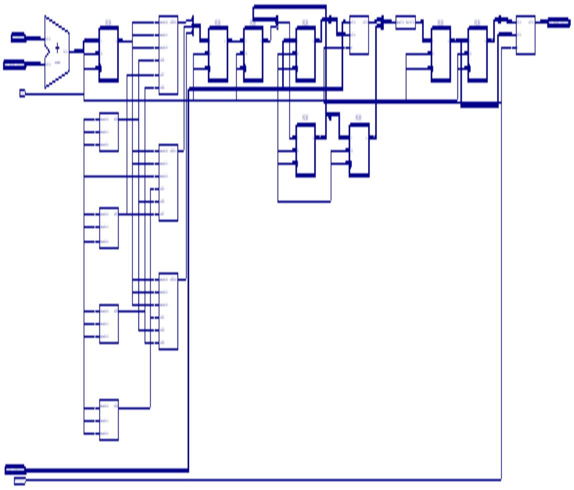

Fig 2: Schematic diagram for Data

5.2 POST SYNTHESIS SIMULATION:

After synthesis, we run post synthesis simulation for verification of our designed functionality. The following table 1 depicts, the usage of operations in implementing the hardware architecture of scalable encryption algorithm. In the proposed architecture design, the number of slice LUT's (look up table) utilization is 44%. It shows advantage in terms of input/output bonds (IOB) which is 59%. IOB provides multiple usage for performance enhancement, through single input. In this paper, the FPGA implementation of hummingbird is shown, asit is applicable for low cost resource constrained devices like RFID tags, smart cards, credit cards and wireless sensor nodes.

Logic	Used	Available	Utilization
Number of Slices:	4135	3584	115%
Number of Slice Flip Flops:	7177	7168	100%
Number of4 input LUTs:	3195	7168	44%
Number of bonded IOBs:	103	173	59%
Number of GCLKs:	1	8	12%

Table 1: Device utilization summary:

5.3 POWER ANALYSIS REPORT

The following tabulation show total power analysis report , When execute VerilogHDL Scalable Encryption Algorithm it provide the power report generated by Project Navigator with Spartan 3 xc3s400 Device .

Power summary:	I(mA)	P(mW)
Total estimated power consumption		245
Vccint 1.20V:	100	120
Vccaux 2.50V:	50	125
Vcco25 2.50V:	0	0
Clocks:	0	0
Inputs:	0	0
Logic:	0	0
Outputs:	0	0
Vcco25	0	0
Signals:	0	0
Quiescent Vccint 1.20V:	100	120
Quiescent Vccaux 2.50V:	50	125

Table 2: Power Analysis

5.4 SIMULATION REPORT

The VerilogHDL File is executed using model sim 6.2 software, we give input values using binary data format that values shows in wave report as follows

Encrypt

InputData: 00011000000101000010000000110000001000 0000010000

InputKey: 00000000001000110010001100100011001000 1100100011

Binary Format

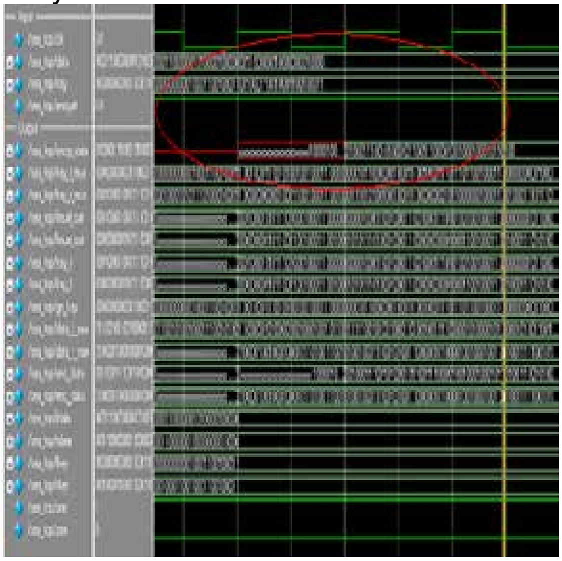

Fig : 3 Encrypted form

Encrypt

Input Data: 00011000000101000010000000110000001000 0000010000

Input Key: 00000000001000110010001100100011001000 1100100011

Encrypted Data : 01000111001010001011001110000100 1000100010110100

Decrypt

Input Data : 01000111001010001011001110000100 1000100010110100 (Output Data From Encrypt)

Input Key : 00000000001000110010001100100011001 0001100100011

Output : 00011000000101000010000000110000001 0000000000010000

Binary Format

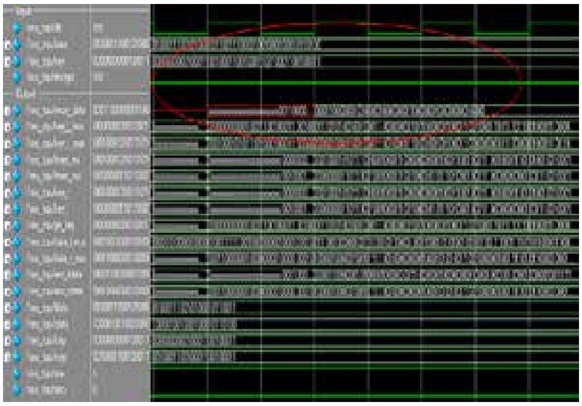

Fig : 4 : Decrypted wave form

6. CONCLUSION

An FPGA implementation of the scalable encryption algorithm based on Spartan3- 3s400ft256-5 of Xilinx Design Suite is presented in this paper. The presented parametric architecture allows keeping the flexibility of the algorithm by taking advantage of generic VHDL coding. It executes one round per clock cycle, computes the round and the key round in parallel and supports both encryption and decryption at a minimal cost. Compared to other recent block ciphers, SEA exhibits a very small area utilization that comes at the cost of a reduced throughput. Consequently, it can be considered as an interesting alternative for constrained environments. The simulation results show that our design The low power and High speed FPGA implementation is very precisely achieved by the proposed algorithm due to its prominent internal structure. Hence this high performance ultra-lightweight hybrid model will meet the power consumption requirements with constricted response time for diverse embedded applications and can be widely suitable for hardware environment.

REFERENCE [1] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, "SEA:A Scalable Encryption Algorithm for Small Embedded Applications," | in the Proceedings of CARDIS 2006, ser. LNCS, vol. 3928, Taragona,Spain, 2006, pp. 222–236. | [2] Data Encryption Standard, NIST Federal Information Processing Standard FIPS 46-1, Jan. 1998. | [3] J. Daemen, V. Rijmen, The Design of Rijndael. Springer-Verlag, 2001. | [4] Advanced Encryption Standard, NIST Federal Information Processing Standard FIPS 197, Nov. 2001. | [5] D. Wheeler and R. Needham, "TEA, a Tiny Encryption Algorithm," in the Proceedings of Fast Software Encryption - FSE 1994, ser. LNCS, | vol. 1008, Leuven, Belgium, Dec. 1994, pp. 363–366. | [6] G. Yuval, "Reinventing the Travois: Encryption/MAC in 30 ROM Bytes," in the Proceedings of Fast Software Encryption - FSE 1997, | ser. LNCS, vol. 1267, Haifa, Israel, Jan. 1997, pp. 205–209. |