International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

FPGA Implementation for Recduced Memory Using Scalable Encryption Algorithm

L. Malathi, L. J. Arthiha, R. Kanmani

***Abstract* – Scalable encryption algorithm (SEA) is a parametric block cipher for resource constrained systems (e.g., sensor networks, RFIDs). 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 and AES Rijndael), 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.**

***Keywords* – AES, DES, FPGA, SEA.**

I. Introduction

Compared to older solutions of low-cost encryption like Tiny Encryption Algorithm (TEA) or Yuval's proposal, 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 Field Programmable Gate Array (FPGA) encryption/decryption core for SEA.

Due to its simplicity constraints, SEA n, b is based on a limited number of elementary operations (selected for their availability in any processing device) denoted as follows: 1) bitwise XOR; 2) addition mod2b; 3) a 3-bit substitution box S:= {0; 5; 6; 7; 4; 3; 1}; that can be applied bitwise to any set of 3-bit words for efficiency purposes.

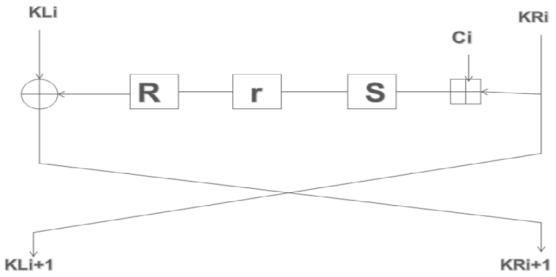

Fig.1.1. Blocking Structure Of Key

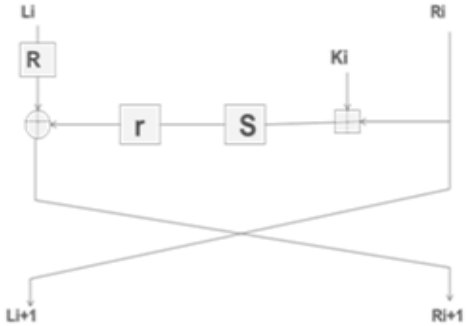

Fig.1.2. Blocking Structure of Encryption

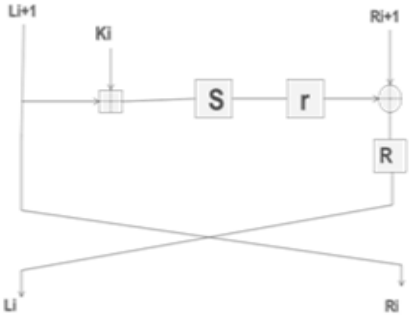

Fig.1.3 Blocking Structure of Decryption Block description
r – Bit rotation,
R – Word rotation,
S – Substitution box,
⊞ - modulo 2 addition,
⊕ - bitwise XOR

II. Round and Key Round

Based on the previous definitions, the encrypt round FE, decrypt round FK are pictured and defined as,

$[Li+1;Ri+1] = FE(Li;Ri;Ki),$

$Ri+1 = R(Li) \text{ xor } r(S(Ri \boxplus Ki))$

$Li+1 = Ri$

$[Li+1; Ri+1] = FD(Li;Ri;Ki),$

$Li = R(Ri+1 \text{ xor } r(S(Li+1 \boxplus Ki)))$

$Ri = Li+1$

$Li+1 = Ri[KLi+1;KRi+1]$

$= FK(KLi;KRi;Ci)\ ,$

$KRi+1 = KLi \text{ xor } R(r(S(KRi \boxplus Ci)))$

$KLi+1 = KRi:$

III. Blocking Structure

The structure of our loop architecture for SEA is depicted in, with the round function on the left part and the key schedule on the right part. Resource-consuming blocks are the S boxes and the mod2b adder; the Word Rotate and Bit Rotate blocks are implemented by swapping wires. According to the specifications, the key schedule contains two multiplexers which allow switching the right and left part of the round key at half the execution of the algorithm using the appropriate command signal Switch. The multiplexers controlled by Half Exec provides the round function with the right part of the round key for the first half of the execution and transmits its left part instead after the switch.

To support both encryption and decryption, we finally added two multiplexers controlled by the Encrypt signal. Supplementary area consumption will be caused by the two routing paths. The algorithm can easily benefit of a modular implementation, taking as only mandatory parameters the size of the plaintexts and keys n and the word length b. The number of rounds nr is an optional input that can be automatically derived from n and b according to the guidelines given in [6].

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

IV. IMPLEMENTATION RESULTS

Implementation results were extracted after place and route with the ISE 8.2i tool from Xilinx on a xc3S500 VIRTEX-4 platform with speed grade 12. In order to illustrate the modularity of our architecture, we ran the design tool for different sets of parameters, with plaintext/key sizes n ranging from 128 to 8 bits and word lengths of and 16 bits. For the control part, we used the recommended number of rounds nr= 3(n=4) +2((n=2b) + (b=2))]. The computed implementation costs stand for both the operative and control parts. A summary of these results is presented in where the area requirements (in slices), the work frequency, and the throughput are provided. We observe that the obtained values for the work frequency are very close for all the implementations. Indeed, the critical path (passing through the key scheduling multiplexers, a mod 2b adder, the Round Function Sbox, a XOR operator and the multiplexer selecting between encryption or decryption paths) is very similar for any of our selected values for n and b. For a given n value, it is noticeable that increasing b decreases the number of rounds nr and, therefore, improves the throughput (since work frequencies are close in all our examples).1+1 if this term is even. Similarly, for our set of parameters, increasing b for a given n generally decreases the area requirements in slices. These observations lead to the empirical conclusion that as long as the b parameter is not a limiting factor for the work frequency, increasing the word size leads to the most efficient implementations for both area and throughput reasons.

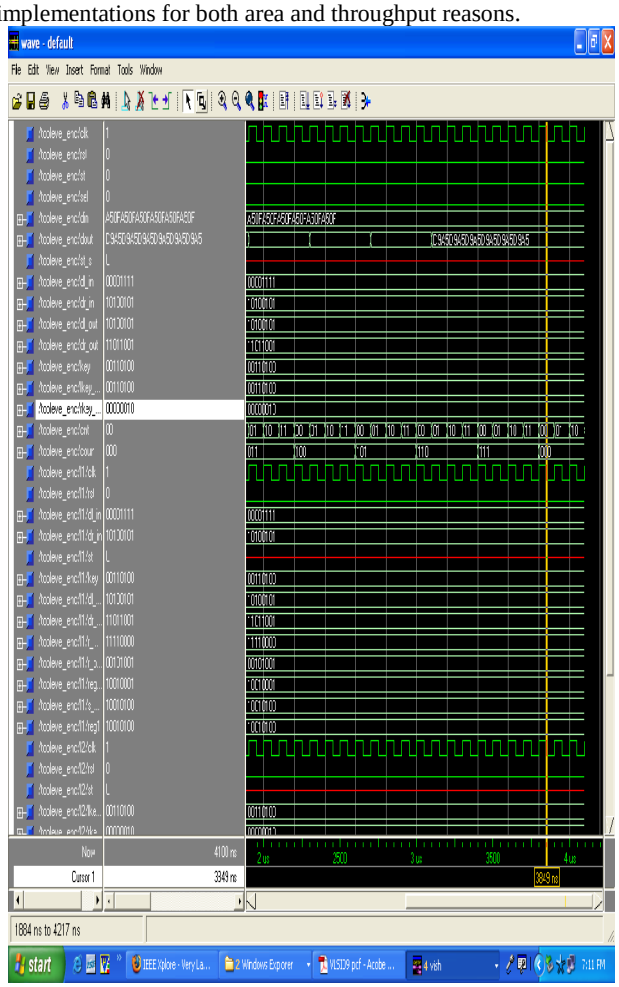

Fig.4.1. Simulation Result for Top Level Encryption

They respectively have no pipeline (Pipe0) or a three-stage pipeline (Pipe3 and use look up table (LUT)-based or distributed RAM-based S boxes. The fifth referenced implementation[5] uses a 32-bit data path and consequently reduces the area requirements at the cost of a smaller throughput. Finally, [7] uses a 128-bit data path with a pipelined composite field description of the S box. As a matter of fact, a lot of other FPGA implementations of the AES can be found in the open literature, e.g., taking advantage of different data path sizes, FPGA RAM blocks, pipelining, unrolling techniques [4].

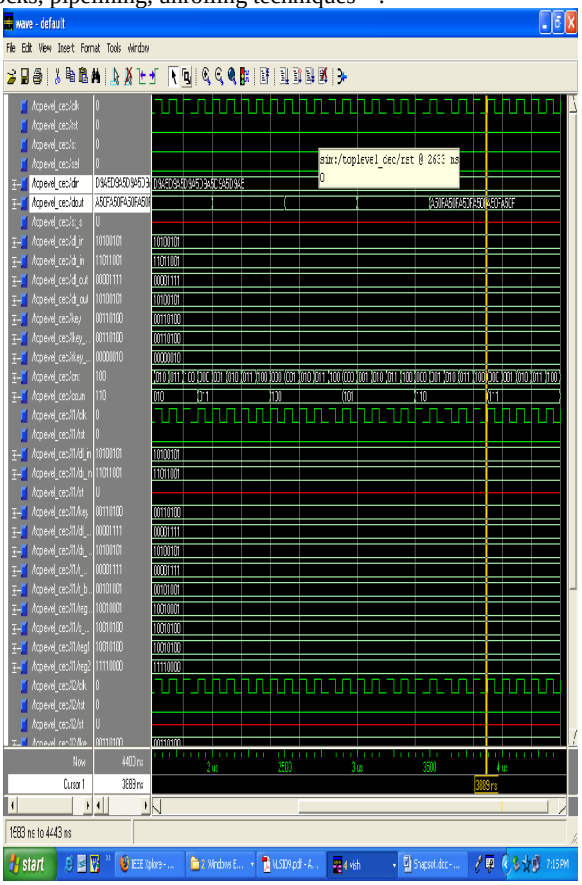

Fig.4.1. Simulation Result for Top Level Decryption

Table 4.1: Implementation of SEA with Different N and B Parameters

N	b	nr	# of slices	# of slice FFs	Output every cycle	Freq (Mhz)	Throughput(Mbps)	Th/Slice
48	4	6	65	74	1/6	278	285	4.3
48	8	3	60	79	1/3	277	284	4.7
72	4	9	80	86	1/9	285	291	3.2
72	6	6	74	90	1/6	283	289	3.9
72	12	3	68	97	1/3	282	288	4.2
96	4	12	85	101	1/12	290	276	3.2
128	8	8	51	86	1/8	274	275	5.3
108	6	9	90	115	1/9	289	283	3.1
126	7	9	95	123	1/9	286	277	2.9
132	11	6	105	133	1/6	292	286	2.7
144	4	18	140	138	1/18	295	289	2.0
144	6	12	130	140	1/12	294	288	2.2
144	8	9	125	149	1/9	293	287	2.2
144	12	6	110	155	1/6	291	285	2.5

Table 4.2: Comparison Results for Other Block Ciphers

Algorithm, Device	nr	Freq (Mhz)	Throughput (MBPS)	Thr / Slice	Bit / slice
AES(Pipe0LUT), Xc2v4000	10	59	760	.277	.047
AES(Pipe0-Dist), Xc2v4000	10	78	1000	.562	.072
AES(Pipe3LUT), Xc2v4000	10	148	1890	.650	.044
AES(Pipe0-Dist), Xcv100e	10	178	2280	1.175	.066
AES, Xc4vlx25	10	215	215		.114
ICEBERG, Xc2v4000	16	988	988	.191	.111
SEA126,7, Xc3S500e	11	156	156	1.718	.302
SEA128,8	7	274	275		.560

V. CONCLUSION

An efficient architecture has been presented FPGA implementations of a Scalable Encryption Algorithm for various sets of parameters. 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.

REFERENCES

[1] Advanced Encryption Standard, FIPS PUB 197, (2001).
[2] Daemen.J and Rijmen.V, (2001)' The Design of Rijndael'New York Springer-Verlag.
[3] Data Encryption Standard, FIPS PUB 46-3(1999).
[4] Gaj.K and Chodowiec.P (2001)' Fast implementation and fair comparison of the final candidates for the advanced encryption standard using field programmable gate array' in proc. Topics Cryptol (CT-RSA) pp.84-99.
[5] Pramstaller.N and Wolkerstorfer.J (2004)' A Universal and efficient AES co-processor for field programmable logic array' in proc. FPL.pp.565-574.
[6] Standaert.F.x, Piert.G, Gershenfeld.N and Quisquater.J.J (2006) 'SEA: A Scalable Encryption Algorithm for small embedded application, pp 222-236.
[7] Standaert.F.X, Rouvroy.G, Quisquater.J.J and Legat.J.D (2003)' Efficient implementation of Rijndael encryption in reconfigurable hardware: improvements and design trade-offs'pp-334-350.
[8] Wheeler.D and Needham .R (1994) 'TEA, a tiny encryption algorithm" pp-363-366.
[9] Yuval.G (1997) 'Reinventing the travois' encryption/ MAC 30 ROM bytes, pp.205-209.
[10] Zambreno.J, Nguyen.D and Choudhary.A (2004)' Exploring are/ delay tradeoffs in AES FPGA implementation pp.75-585.

AUTHOR'S PROFILE

L. Malathi
received the BE degree in Electronics and Communication Engineering from Christian College of Engineering and Technology, Anna University Chennai in 2005 and the ME degree in Applied Electronics from PSNA College of Engineering and Technology, Anna University Coimbatore in 2008. Currently working in Sri Ramakrishna Institute of Technology as an Assistant Professor in ECE Department. She has five years teaching experience. She has presented papers in National Conference. Under her guidance, the final year students are doing their project in various fields. Her field of interest is VLSI. Email: lmalathigraj@gmail.com

L. J. Arthiha
received the B.Tech. degree in Electronics and Communication Engineering from SASTRA University in 2009 and the M.Tech degree in VLSI Design from SASTRA University in 2011. Currently working in Sri Ramakrishna Institute of Technology as an Assistant Professor in ECE Department. She has one year teaching experience. Under her guidance, the final year students are doing their project in various fields. Her field of interest is VLSI.
Email: arthihalj@gmail.com

R. Kanmani
received the BE degree in Electronics and Communication Engineering from P.R Engineering College, Anna University Chennai in 2007 and the M.Tech degree in Advanced Communication Systems from SASTRA University in 2009. Currently working in Sri Ramakrishna Institute of Technology as a Lecturer in ECE Department. She has three years teaching experience. She has presented papers in National and International Conference. Under her guidance, the final year students are doing their project in various fields. Her field of interest is Communication Systems.
Email: kanmanivan@gmail.com