Available online at www.sciencedirect.com

ScienceDirect

Journal of Electrical Systems and Information Technology 2 (2015) 178–183

www.elsevier.com/locate/jesit

Power efficient and high performance VLSI architecture for AES algorithm

K. Kalaiselvi [a,*], H. Mangalam [b]

[a] *Department of Electronics and Communication Engineering, Hindusthan College of Engineering and Technology, Coimbatore, India*

[b] *Department of Electronics and Communication Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, India*

Received 9 November 2014; received in revised form 18 March 2015; accepted 25 April 2015
Available online 11 September 2015

Abstract

Advanced encryption standard (AES) algorithm has been widely deployed in cryptographic applications. This work proposes a low power and high throughput implementation of AES algorithm using key expansion approach. We minimize the power consumption and critical path delay using the proposed high performance architecture. It supports both encryption and decryption using 256-bit keys with a throughput of 0.06 Gbps. The VHDL language is utilized for simulating the design and an FPGA chip has been used for the hardware implementations. Experimental results reveal that the proposed AES architectures offer superior performance than the existing VLSI architectures in terms of power, throughput and critical path delay.

Keywords: Advanced encryption standard (AES) algorithm; VLSI architecture; Low power; FPGA implementation

1. Introduction

In the recent years, there is a growing requirement to implement cryptographic algorithms in fast rising high-speed network applications. Encryption is the process of encoding information so that the unauthorized persons cannot identify the information. All the encryption algorithms convert the available information into unreadable secured form, referred to as cipher text. The authorized person will be able to decode the information using decryption algorithms. Two types of cryptographic systems available for data security are asymmetric (public-key) and symmetric (secret-key) cryptographies (Hosseinkhani and Javadi, 2012). Asymmetric cryptography utilizes separate keys for encryption and decryption process for the key transportation mechanism. Conversely, symmetric cryptography utilizes an identical key for both encryption and decryption process, which is effective while handling a large amount of data (Chen et al., 2011).

[*] Corresponding author. Tel.: +91 7598254742.
E-mail address: kkalaiselvihind@gmail.com (K. Kalaiselvi).
Peer review under responsibility of Electronics Research Institute (ERI).

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jesit.2015.04.002

Advanced encryption standard (AES) has been recognized as an efficient scheme for VLSI implementation comparing with available symmetric cryptographies (Standard, 2001). Modern cryptographic application specific integrated circuits (ASICs) and coprocessors are fabricated using AES algorithm, so that it will become one of the most key symmetric ciphers in the coming years (Aes, 2001). The AES algorithm is a modified version of the Rijndael cipher and can be implemented efficiently in both hardware and software (Daemen and Rijmen, 2002). The AES algorithm could be implemented using high performance modern microprocessors to satisfy high throughput requirements in most applications. But the power optimization issues cannot be addressed in these processors. In view of power, speed and area requirements, VLSI based hardware description language (HDL) implementation of the AES algorithm is important for low power consumption and small silicon area.

AES encryption systems were proposed in the last decade, based on field programmable gate array (FPGA) (Daemen and Rijmen, 2002; Tillich et al., 2005; Zhang and Parhi, 2004; Yicheng et al., 2008) and ASIC (Rodriguez-Henriquez et al., 2003; Jing et al., 2007; Rahimunnisa et al., 2014) hardware. Both the FPGA and ASIC implementations provide more physical security, but the speed, power consumption and area utilization are different in these approaches. Loop-unrolling, Pipelining and Cell array hardware architecture are some of the techniques used for AES algorithm implementation. The pipelining architectures are used to improve the speed of the implementation and achieve high throughput (Verbauwhede et al., 2003). Sub-pipelined architectures have been used in the literature for minimizing area and provide best speed/area ratio (Verbauwhede et al., 2003; Hodjat and Verbauwhede, 2006). However, reconfigurable hardware system using cell array architecture is helpful in efficient AES implementation (Li et al., 2012). The reconfigurable array architectures are applied for high-reliability designs in communication applications and biomedical signal processing applications (El-Rayis et al., 2008; Milovanović et al., 2009; Satheeskumaran and Sabrigiriraj, 2014).

In this paper, we focus on reconfigurable hardware implementation of the novel AES S-box in a uniform and coherent way using key expansion approach. The chosen performance metrics are throughput, the critical path delay and power consumption. These metrics are essential for performance analysis in FPGA devices. The amicable solution using reconfigurable implementation is a highly promising alternative and could achieve superior performances. For design implementations, the very high speed integrated circuit hardware description language (VHDL) was used. Simulation, synthesis and implementation are performed using Xilinx integrated software environment (ISE) tool and Modelsim. The critical path time, area requirement and power consumption are also analyzed using Xilinx.

2. VLSI implementation of AES algorithm

AES algorithm supported cell array reconfigurable has been proposed for the system versatile function to implement high reliableness and high performance (Chodowiec, 2002). The cell array reconfigurability is combined with state structures to implement the proposed key expansion scheme. The State structure of AES will support high throughput cipher engine. The AES algorithm steps are performed on a two-dimensional cell array of bytes known as the State. Every cell unit bit width of state is assigned to 8-bit so the array process 256-bit encryption data in each spherical. The four bytes in every column of the state array form 32-bit words, wherever the row provides an index for the four bytes at intervals every word (Li et al., 2012). Fig. 1 shows the block diagram of the cell unit.

3. NIST and DOR schemes

The NIST scheme is capable of completing 64, 128, 192 and 256 bit key expansions for a more secure cryptography method. These variations do not create a large impact. It has an effect on the throughput per slice metric because of the accumulated hardware needs. Shared hardware using multiplexers is to route the information within the correct path. It is essential to notice that Chodowiec scheme did not include a key schedule module and key expansion hardware (Chodowiec, 2002). The increase in encryption size cannot be handled using this scheme. Various implementations on a similar device should be analyzed in depth to justify the encryption size of Chodowiec scheme.

The Direct Optimized Routing (DOR) Scheme is based on 128 bit data path width, which could be partitioned into narrower data paths based on the requirements. The data transfer takes place through three 128 bit data buses; one for data bus, one for the key, and one for the encrypted output (Van Dyken and Delgado-Frias, 2010). The repeated round block pattern can be used as the open core scheme for DOR scheme implementation. Each single round would take

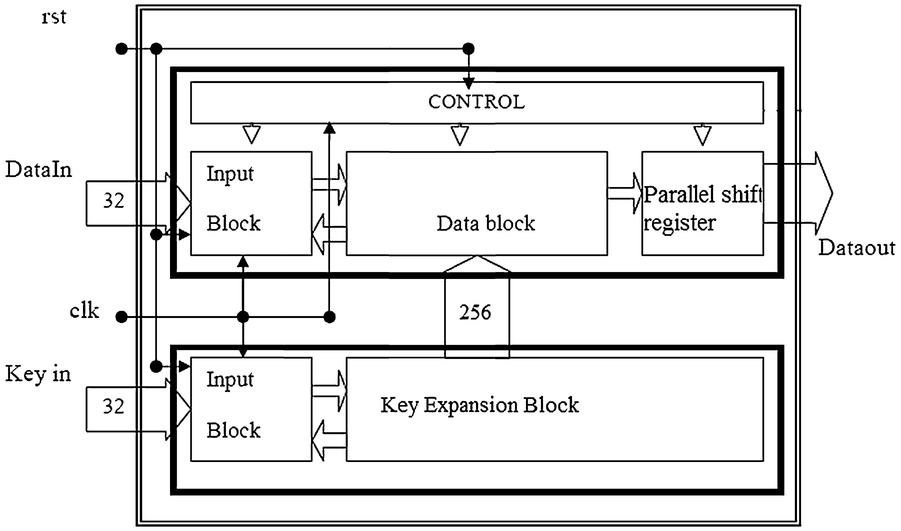


Fig. 1. AES algorithm implementation.

only one clock cycle for data transfer. The DOR scheme is usually implemented with an extra spherical block and key expansion unit that stores all round keys in a read only memory (ROM).

4. Proposed method

The proposed work is based on the key expansion with dual stage design. The dual stage scheme is used to determine the effect of multiple round blocks in power consumption. Most of the high speed designs that utilize loop unrolling and pipelining aim to increase throughput (Van Dyken and Delgado-Frias, 2010). Since our technique is able to complete the key expansion of 64 bit keys internally, the control logic to implement concurrent encryptions would effect a further round block on power consumption due to the dynamic design. Fig. 2 shows the block diagram of the dual stage scheme. The existing DOR scheme consumes eleven clock cycles to complete the single round and the encrypted data

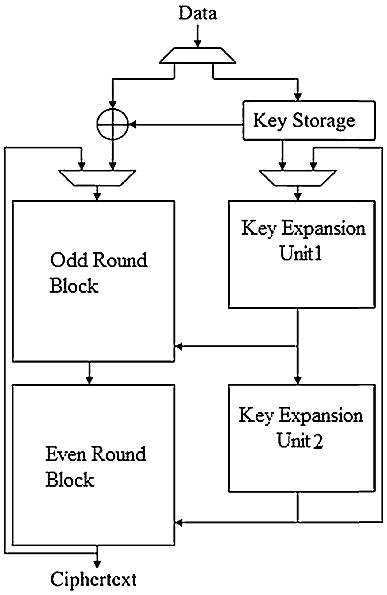


Fig. 2. Proposed implementation with dual stage scheme.

is sent through output data bus. However the dual stage in the proposed design takes only six cycles for performing each encryption; five cycles being used in encryption process and sixth cycle for data output.

Most of the encryption schemes are capable enough to perform the key expansion of 128 bit keys internally barring the Gaj scheme (Chodowiec, 2002). The NIST scheme is capable of completing 192 and 256 bit key expansions for implementing a more secure encryption process. The actual throughput of the system may be affected due to the increased hardware requirements. Shared hardware using multiplexers has been utilized for routing the data in proper path. It is significant to note that Gaj scheme did not include a key schedule module and key expansion hardware. For speed and power optimization, Virtex ball grid array package FPGA chip is used for implementation. Once the ROM was implemented utilizing FPGA memory there is a reduction of LUT slices comparing to the DOR scheme. By combining encryption and decryption, the area efficiency and throughput performance improves.

5. Results and discussion

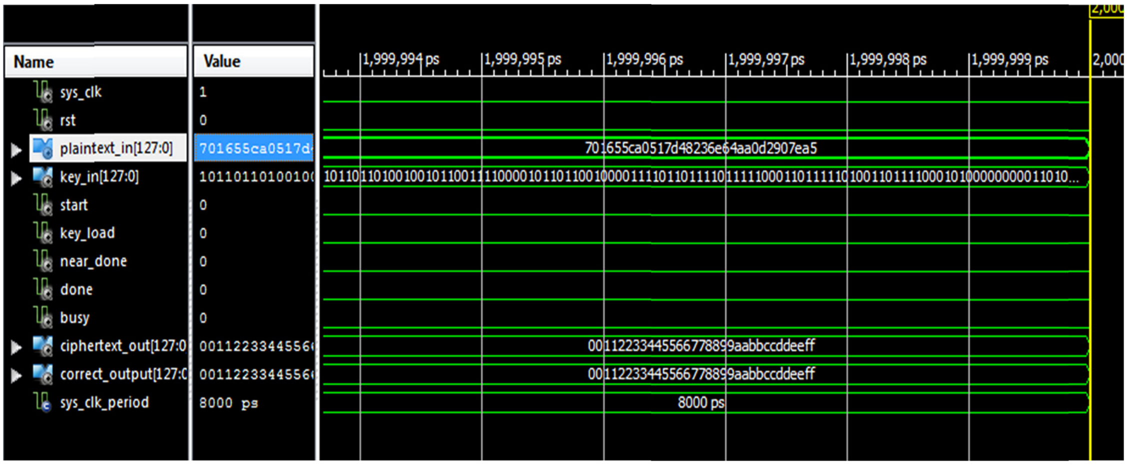
Xilinx ISE tool has been used for the synthesis, place-and-route, and timing analysis. The pipelined implementation of the proposed design, we reach a clock cycle of 3.6182 ns (277.4 MHz of operative frequency). Once the function of individual modules is verified for correctness, these can be clubbed together. To support the above-mentioned approach, the crypto algorithm is split into two modules: coding and secret writing. The synthesis of the chip is performed within the XILINX tool targeting Xilinx Virtex 5 technology (XC5VLX30 target device) and therefore the report is given in Table 1. The combination of Modelsim and Xilinx design flow has been used for the entire process. Individual register transfer logic (RTL) is obtained once synthesizing the VHDL style. The timing simulation is additionally performed to verify the functional correctness of the planning. However, the RTL diagram is not enclosed here for conciseness.

The power analysis is performed using Xilinx's XPower analysis tool. The Virtex 5 Pro is a target device, as it is a full featured and flexible FPGA that contains two Power PC cores and plenty of logic cells and I/O pin counts ranging from 208 to 1164 (Xilinx). It is set to run at 25 MHz during the simulations. Fig. 3 shows the simulation result for the encryption of a test vector. The design is synthesized in Xilinx Environment. The target device is xc5vlx30 in the family of Virtex5.

Through the analysis of the schemes for throughput and power, it is evident that the proposed scheme outperforms the existing schemes. The standard national institute of standards and technology (NIST) and direct optimized routing (DOR) methodologies have been compared with the proposed technique. Table 2 compares the critical path, throughput, and power of the proposed technique with NIST and DOR techniques. The throughput of the DOR scheme is more

Table 1
Synthesis report for the proposed scheme.

Device utilization summary	
Slice logic utilization	
Number of Slice Registers	5493
Number of Slice LUTs	22,659
Slice logic distribution	
Number of Bit Slices used	23,424
Number with an unused Flip Flop	17,931
Number with an unused LUT	765
Number of fully used Bit Slices	4728
IO utilization	
Number of IOs	400
Specific feature utilization	
Number of BUFG/BUFGCTRLs	2
Timing summary	
Minimum period	3.6182 ns
Maximum frequency	277.4 MHz
Minimum input arrival time before clock	2.645 ns
Maximum output required time after clock	3.420 ns



Name	Value	
sys_clk	1	
rst	0	
plaintext_in[127:0]	701655ca0517d	701655ca0517d48236e64aa0d2907ea5
key_in[127:0]	1011011010010(	1011011010010010110011110000101101100100001111011011110111110001101111101001101111000101000000000011010...
start	0	
key_load	0	
near_done	0	
done	0	
busy	0	
ciphertext_out[127:0	00112233445566	00112233445566778899aabbccddeeff
correct_output[127:0	00112233445566	00112233445566778899aabbccddeeff
sys_clk_period	8000 ps	8000 ps

Fig. 3. Encryption of a test vector.

Table 2
Comparison of critical path, throughput and power.

Scheme	Critical path time (ns)	Throughput (Mbps) with No overhead	No. of 4 input LUTs	Total power (mW)
NIST	5.68	175.8	515	47.94
DOR	4.62	216.3	1152	23.33
Proposed	3.63	277.4	351	13.21

comparing to the NIST scheme at the cost double number of required LUT slices for implementation. However, the proposed scheme provides the highest throughput of 277.4 Mbps with 31.8% reduction respectively in the LUT slices.

6. Conclusion

From the obtained results, it is evident that the proposed scheme could be able to operate at higher clock frequencies than the existing schemes. The DOR technique with the throughput of 216.3 Mbps could be able to reduce the logic and signal power requirements. But the proposed work reduces the power requirement by as much as 43.4% with critical path time is reduced to 21.4%. Furthermore, newer FPGAs will offer even greater performance improvement.

References

Aes, N.I.S.T., 2001. Advanced Encryption Standard. Federal Information Processing Standard, FIPS-197, pp. 12.

Chen, R.J., Lin, J.J., Hung, S.M., Lai, J.L., Horng, S.J., 2011. Architecture design of high-efficient and non-memory AES crypto-core for WPAN. Concurr. Comput.: Pract. Exp. 23 (12), 1332–1347.

Chodowiec, P.R., (Doctoral dissertation) 2002. Comparison of the hardware performance of the AES candidates using reconfigurable hardware. George Mason University.

Daemen, J., Rijmen, V., 2002. The Design of Rijndael. Information Security and Cryptography. Text and Monographs. Springer Verlag.

El-Rayis, A.O., Arslan, T., Erdogan, A.T.,2008. Addressing future space challenges using reconfigurable instruction cell based architectures. In: Adaptive Hardware and Systems, 2008. AHS'08. NASA/ESA Conference on, June 22–25. IEEE, Noordwijk, pp. 199–203.

Hodjat, A., Verbauwhede, I., 2006. Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors. IEEE Trans. Comp. 55 (4), 366–372.

Hosseinkhani, R., Javadi, H.H.S., 2012. Using cipher key to generate dynamic S-Box in AES cipher system. Int. J. Comp. Sci. Security (IJCSS) 6 (1), 19–28.

Jing, M.H., Chen, Z.H., Chen, J.H., Chen, Y.H., 2007. Reconfigurable system for high-speed and diversified AES using FPGA. Microprocess. Microsyst. 31 (2), 94–102.

Li, H., Ding, J., Pan, Y., 2012. Cell array reconfigurable architecture for high-efficiency AES system. Microelectron. Reliab. 52 (11), 2829–2836.

Milovanović, E.I., Nikolić, T.R., Stojčev, M.K., Milovanović, I.Ž., 2009. Multi-functional systolic array with reconfigurable micro-power processing elements. Microelectron. Reliab. 49 (7), 813–820.

Rahimunnisa, K., Karthigaikumar, P., Kirubavathy, J., Jayakumar, J., Kumar, S.S., 2014. A 0.13-μm implementation of 5 Gb/s and 3-mW folded parallel architecture for AES algorithm. Int. J. Electron. 101 (2), 182–193.

Rodriguez-Henriquez, F., Saqib, N.A., Diaz-Perez, A., 2003. 4.2 Gbit/s single-chip FPGA implementation of AES algorithm. Electron. Lett. 39 (15), 1115–1116.

Satheeskumaran, S., Sabrigiriraj, M., 2014. A new LMS based noise removal and DWT based R-peak detection in ECG signal for biotelemetry applications. Natl. Acad. Sci. Lett. 37 (4), 341–349.

Standard, N.F., 2001. Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication, pp. 199.

Tillich, S., Großschädl, J., Szekely, A., 2005 January. An instruction set extension for fast and memory-efficient AES implementation. In: Communications and Multimedia Security. Springer, Berlin, Heidelberg, pp. 11–21.

Van Dyken, J., Delgado-Frias, J.G., 2010. FPGA schemes for minimizing the power-throughput trade-off in executing the Advanced Encryption Standard algorithm. J. Syst. Architect. 56 (2), 116–123.

Verbauwhede, I., Schaumont, P., Kuo, H., 2003. Design and performance testing of a 2.29-GB/s Rijndael processor. IEEE J. Solid-State Circuits 38 (3), 569–572.

Xilinx, Inc., Xilinx Virtex-II Pro FPGAs. http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex_ii_pro_fpgas/index.htm (accessed 1.10.07).

Yicheng, C., Xuecheng, Z., Zhenglin, L., Yu, H., Zhaoxia, Z., 2008. Energy-efficient and security-optimized AES hardware design for ubiquitous computing. J. Syst. Eng. Electron. 19 (4), 652–658.

Zhang, X., Parhi, K.K., 2004. High-speed VLSI architectures for the AES algorithm. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 12 (9), 957–967.