An Implementation of Configurable and Small-Area AES IP Core Oriented Avalon Bus

X.C. Tao
Institution of VSLI Design, Hefei University of Technology
China

D.L. Zhang
Institution of VSLI Design, Hefei University of Technology
China

Y.K. Song
Institution of VSLI Design, Hefei University of Technology
China

Abstract—**The Advanced Encryption Standard (AES) issued by the National Institute of Standards and Technology in 2001 has become the new widely-used symmetric block cipher standard. A lot of efforts have been made on the various hardware implementations of the AES algorithm. Some focus on achieving low-cost constructions, while others focus on designing high throughput. Given the specific requirement of wireless communication and portable devices, this article presents an AES IP core with an acceptable trade-off between performance and area. By introducing composite fields Sboxes and researching optimization of MixColumn can reduce resources. The AES IP is designed based on Avalon bus. It is compatible with five modes including ECB, CBC, OFB, CFB, and CTR. 128,196,256 bits key are also supported. Meanwhile, it can be flexibly configured according to the specific circumstances. This design and implementation of the AES core has a certain value for the generalization the wireless communication terminal hardware platform.**

Keywords-AES; configurable; SoC; Avalon

I. INTRODUCTION

Encryption has become an essential component for secure data and telecommunications networks. It makes electronic commerce, payment systems and transactions over networks possible. It has become one of the main tools for privacy, trust, access control, corporate security and countless other areas. AES Algorithm is safe, efficient and high-performance. Hence, there is an increasing need of hardware implementation of AES algorithm.

Research efforts have been made for the implementation of AES security algorithms on hardware for embedded system applications using FPGA. In the designs [2] [3], the core is designed with Microblaze processor core. [4] [5] use Altera NIOS II core. In this paper, we propose an Advanced Encryption Standard (AES) based on Altera NIOS II core, implement the design and simulation of AES module compatible with five modes based on Avalon bus. At the same time, AES module, NIOS II core and peripherals have been integrated with SOPC technology. The whole design has a simple hard structure, and it is flexibility and safety. It implements an efficient, cost-effective solution. In this paper, our specific contributions are:

- Design of AES module compatible with five modes (including ECB(Electronic codebook) CBC(Cipher-block chaining) OFB(Output feedback) CFB(Cipher feedback) CTR(Counter)).
- The AES module supports 128,196,256 bit key.
- The design is optimized in terms of resources.
- The design based on Avalon bus can be configured by slave port and the mutual communication for data and key exchanges are achieved by a suitable implementation of the Avalon bus.

The rest of the paper organized as follows. A brief introduction to AES algorithm is stated in Section II. In Section III, we discuss on the algorithm optimization of AES. Section IV gives a description of the proposed system architecture and its major building blocks. Conclusions are made in Section V.

II. ADVANCED ENCRYPTION STANDARD

The AES is an iterative, symmetric block algorithm with input packet size of 128 bits and key lengths of 128, 192, and 256 bits. The Algorithm per round operates on the intermediate results, called state, which is a rectangular array of bytes. Since the block size is 128 bits, the rectangular array is of dimensions 4x4. Similarly, the round key, which is expanded from the initial key, can be imagined as a rectangular array.

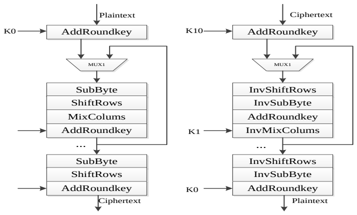

FIGURE I. MIXCOLUMN/INVMIXCOLUMN.

Figure.I describes the AES algorithm with 128bit key. There are different steps in each round of the encryption/ decryption data path. The individual transformations are Subbyte, Shiftrow, Mix-column, and Addroundkey.

III. Algorithm Optimization of AES

A. Optimization of S-Box

The single nonlinear step is the SubBytes step, which operates independently on each byte of the state using a substitution table (S-box). This nonlinear function involves two transformations, including taking the multiplicative inverse in the Galois field GF(28) and applying affine transformation. The Inverse is a complex calculation, so that a LUT is used in many current implementations.

But for hardware implementations of AES algorithm, one main drawback of the LUT approach to the S-box occurs: each copy of the LUT requires 256×1 bytes of storage. [6] [7] all put forward some implementations of S-box with combinational logic gate. The main idea is reduced order of Galois Fields. After considering the methods from the reference, we choose this solution that the multiplicative inversion in GF(28) being carried out in GF((24)2).

The multiplicative inversion in GF(28) can be carried out in GF(24) by the architecture by the architecture illustrated in Figure. II ×represents multiplication, x-1 represents inversion in Galois field GF(24). That element is mapped to its composite field representation via isomorphic function δ from GF(28) to GF(24), and mapped back via its inverse function δ-1.

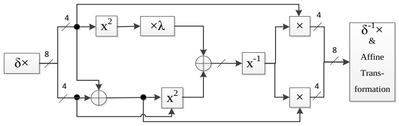

FIGURE II. MIXCOLUMN/INVMIXCOLUMN.

Isomorphic mapping can be described as a matrix multiplication $\delta(x) = T \bullet x$. [7] proposed an algorithm of searching T matrix with $\lambda = \{1110\}$.

B. Implementation of Mixcolumn

In the step of mixcolumn/invmixcolumn, two operations are defined as following.

$$\mathrm{out}_x = [2\ 3\ 1\ 1] \bullet [a\ b\ c\ d]^T \quad \mathrm{out}_y = [E\ B\ D\ 9] \bullet [a\ b\ c\ d]^T$$

[8] Put forward optimization of mixcolumn/invmixcolumn. $03 = 02 + 01$, $09 = 08 + 01$, $0B = 09 + 02$, $0D = 0B + 02$, $0E = 0D + 01$. Only the products of the byte with 01, 02, 04 and 08 are need to be calculated. All elements of the matrices can be formed by summing these products. For $b \in GF(2^8)$ that is expressed by b7b6b5b4b3b2b1b0. ×2 will be b6b5b4 (b3^ b7) (b2^ b7) b1(b0^ b7) b7 which is showed in Fig.4. In this case, ×4 and ×8 can be figured out based on ×2.

$$\times 4 = b_5 b_4 (b_3 \hat{} b_7)(b_2 \hat{} b_7 \hat{} b_6)(b_1 \hat{} b_6)(b_0 \hat{} b_7)(b_7 \hat{} b_6)\ b_6$$

$$\times 8 = b_4 (b_3 \hat{} b_7)(b_2 \hat{} b_7 \hat{} b_6)(b_1 \hat{} b_6 \hat{} b_5)(b_0 \hat{} b_7 \hat{} b_5)(b_7 \hat{} b_6)(b_6 \hat{} b_5) b_5$$

IV. System Architecture

In this section we introduce the system design of the prototyping system and main architectures of AES. Nios II processor is used to integrate various peripherals, while SOPC builder is used to connect Nios II processor and the peripherals. The content of this section is divided into three parts: configurable architectures of AES, the structure of main operation module and system architecture.

A. Configurable Architectures of AES

The AES module designed in this paper is compatible with five modes, including ECB, CBC, CFB, OFB and CTR. The main structure of the model is illustrated by the following image. It consists of the following sections: two main registers, logical operators of encryption/decryption, a number of switch MUX and XOR.

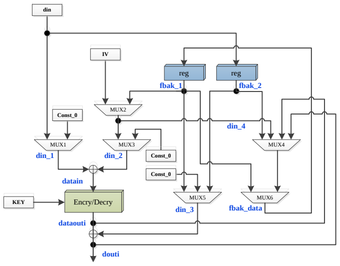

FIGURE III ARCHITECTURES OF AES.

B. Construction of AES Core

The following figure shows a main construction of AES core. The datapath of the core is 128 bit. The main modules are: AES arithmetic unit, key expansion unit, control unit, memory and invkey module. Key expansion unit is to expand encryption key. AES arithmetic unit is to complete the encryption or decryption operation. Control unit is to generate round signals, effective signals, enable signal and so on. The invkey module is used to "invert" the decryption key. The scheme shown in the figure requires a 60 x 32 memory. The AES core works on the principle of the "Equivalent Inverse Cipher" [9]. A memory is only needed during decryption actually. No storage is required for encryption since keys can be expanded on the fly.

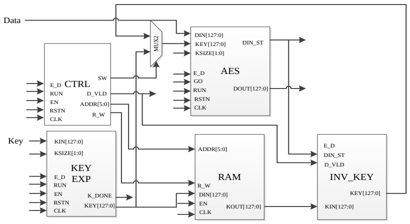

FIGURE IV. CONSTRUCTION OF AES.

C. System Design of AES Core

Now we introduce the system design of AES. We use Avalon bus as the interface of AES core. Burst mode transmission of read and basic mode transmission of write is used. The figure below shows the signal relationship between AES core and Avalon bus.

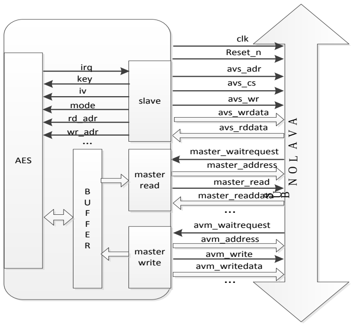

FIGURE V. CONSTRUCTION OF AES BASED ON AVALON.

There are an interrupt and three channels in the design, including configuration channel, read channel and write channel.

Configuration channel : Avalon slave interface is used to configure registers of AES core, including an initial key, IV, mode, feedback size, number of operation, reading/writing address, start flag and interrupt identification.

Read channel : After the initialization and configure all the registers. Write "1" to start flag register, then AES core will read data from the reading address set through configuration channel. Since the datapath of Avalon master interface is 32 bits, but the AES core' datapath is 128 bits, so shifting registers are used to transform data width.

Write channel : Shifting registers are used to transform data width(from 128 to 32). When there is result from AES arithmetical unit, 32bits data is output to the writing address configured.

IRQ : Once the input data number reachs number of operation configured, IRQ is generated. Meanwhile, the Avalon master read interface of AES core stops working. IRQ register should be cleared when starting a new operation.

D. System Structure of Verification

The system structure of verification is described as follows. 264 codec, video capture and wireless transmission are used in this system. A Digital camera is used to data collection. After that, video coding is done. Data from wireless sender is encrypted by AES core. In the wireless receiver, the data received is first decrypted, then video decoded. Finally, image is displayed on the screen.

V. ANALYSIS OF RESULTS AND CONCLUSIONS

We use Verilog HDL to achieve AES core, and verifies them on Cyclone V FPGA. All test schemes are based on Nios II standard core. Synthesis result and performance between those from references is shown in Table 1. The system architecture is shown in Fig. Ⅵ. Two boards are communicated via wireless. Decrypted image is displayed on the screen to verify whether the AES encryption/decryption procedure is done successfully. After testing with system clock set 64Mkhz, the system work stable. We can see intact image consecutively.

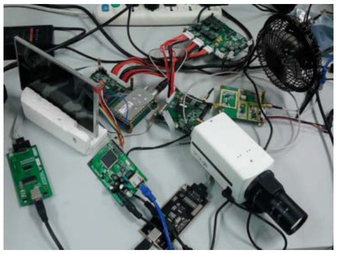

FIGURE VI. SYSTEM PROTOTYPE.

The implementation in [2] has reduced the circuit area to a few hundred slices at the expanse of lowering speed, but the datapath is 32-bit, and is not based on Avalon bus architecture. [7] supports 128,196,256 bit key with normal performance. In design [3], it achieves low-cost, but it doesn't support all 5 kinds of mode and have lower throughput than our design. Design proposed in [4] achieves higher throughput with respect to our design, but it takes much more resources than our design. The implementation in [5] is based on co-processor architecture, but it does not support all 5 modes and has lower throughput.

The AES core in our design which can be flexibly configured according to the specific circumstances is full-featured with moderate performance. Meanwhile, it can enhance the integration of system with standard Avalon bus interface. The design has high value in practical application.

TABLE Ⅰ. SYNTHESIS RESULTS AND COMPARISON.

Work-device	***Combinational ALUTs***	***F max (MHz)***	***Throughput (Mbps)***	***Efficiency (Mbps/Mhz)***
Cyclone-V[our]	4620ALUTs	112	1303/1102/956	11.6/9.8/8.5
ASIC module[7]	22610 Gates	100	188	1.88
Spartan-3[2]	110slices	75	218	2.9
Virtex-5[3]	1774slices	125	1067	8.5
FPGA[4]	16519 logic cells	100	1850	18.5
EP1C4[5]	2144LE	137.46	800	5.82

ACKNOWLEDGMENT

Support for this work was provided by The National Natural Science Foundation of China (61204024, 61179036, 61106020) and Natural Science Foundation of Jiangsu Province of China (BK2011185)

REFERENCES

[1] P. MacKenzie and M. Reiter. Networked cryptographic devices resilient to capture. InSecurity and Privacy, 2001. S P 2001. Proceedings. 2001 IEEE Symposium on, pages 12 –25, 2001.

[2] Chang K H, Chen Y C, Hsieh C C, et al. Embedded a low area 32-bit AES for image encryption/decryption application[C], Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on. IEEE, 2009: 1922-1925..

[3] Sau S, Paul R, Biswas T, et al. A novel AES-256 implementation on FPGA using co-processor based architecture[C], Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ACM, 2012: 632-638.

[4] Biglari M, Qasemi E, Pourmohseni B. Maestro: A high performance AES encryption/decryption system[C], Computer Architecture and Digital Systems (CADS), 2013 17th CSI International Symposium on. IEEE, 2013: 145-148.WS

[5] Feng B, Qi D, Haiwen H. Parallel and multiplex architecture of AES-CCM coprocessor Implementation for IEEE 802.15. 4[C], Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on. IEEE, 2013: 149-153.

[6] Canright.D. A Very Compact S-Box for AES. In: Rao, J., Sunar, B. (eds.) CHES2005. LNCS, vol. 3659, pp. 441–455. Springer, Heidelberg (2005)

[7] Hanjia liu. The design and verification of AES encryption algorithm IP core [D][D]. Shanghai Jiao Tong University, 2009.

[8] Li H, Friggstad Z. An efficient architecture for the AES mix columns operation[C], ISCAS (5). 2005: 4637-4640.

[9] Standard N F. Announcing the Advanced Encryption Standard (AES)[J]. Federal Information Processing Standards Publication, 2001, 197.