Mixed Bus Width Architecture for Low Cost AES VLSI Design

Yibo FAN*, Jidong WANG, Takeshi IKENAGA, Satoshi GOTO
Graduate School of Information, Production and Systems
Waseda University, Japan
*Email: fanyibo@ruri.waseda.jp

Abstract
With the increase of security problem, AES is widely used in a lot of secure systems. For some low data throughput applications, low cost design is more attractive than high speed design. In this paper, low cost hardware architecture for AES algorithm is proposed. Mixed bus width architecture is used to reduce hardware cost and shorten critical path. The experimental results show that the lowest hardware cost implementation of AES algorithm is 4678 Gates. The corresponding frequency is 80MHz and the throughput is 51Mbps. This architecture is very suitable for mid-throughput, low power and low hardware cost systems such as mobile system.

1. Introduction

In the recent years, internet and communication technology are developed very quickly. The security problem becomes more and more important while exchanging information through these media. To protect the important data in unsecure transmission channel, the cryptographies are widely used. There are two major kinds of cryptographies: symmetric cryptography and asymmetric cryptography. Symmetric cryptography is used to encrypt/decrypt large size of data, while asymmetric cryptography is used to do authentication, digital signature and so on.

Advanced Encryption Standard (AES) [1] was selected by the National Institutes of Standards and Technology (NIST) as a new encryption standard to replace the Data Encryption Standard (DES) in Oct. 2000. A lot of hardware architectures for AES algorithm have been proposed. There are two trends of design: The first one is high performance design, which cost a lot of hardware resource. The second one is low cost design.

This paper focuses on low cost implementation of AES algorithm. There are a lot of low cost designs that already have been proposed. Some focus on architecture design, and the most famous one is Satoh's work in [2]. Some focus on S-Box design, and the most important work has been done by Canright in [3]. Some focus on ultra low cost AES design which can be used in RFID, which can be found in [4]. The throughput of low cost designs is much lower than high performance designs. Their throughput can be from tens of kbps [4] to hundreds of Mbps [2, 5-6].

In this paper, a new low cost AES architecture is proposed. This architecture is designed for middle throughput, low power and low cost system such as video encryption system. The proposed architecture in this paper is different from all of the existing architectures. It uses mixed bus width (8-bit bus for S-Box, 32-bit bus for MixColumns). The advantage of this architecture includes two points: 1) Low hardware cost. Since S-Box costs a lot of hardware, only one S-Box is used in this architecture. 2) Short critical path. The proposed architecture is also a parallel architecture. The critical path is shorter than the serial data path design.

This paper is organized as follows. AES algorithm is introduced in Section 2. The mixed bus width architecture is presented in Section 3. The experimental results and comparison are given in Section 4. Finally, conclusion is provided in Section 5.

2. AES Algorithm

AES, also known as Rijndael, is the most popular algorithm used in symmetric key cryptography. AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. AES operates on a 4×4 array of bytes termed the *State*. For encryption, it implements a round function 10, 12, 14 times (depends on the key length).

Most of the AES designs use decryption flow to implement AES into hardware as shown in Figure 1. And in each round of AES (except the first and the last round), it consists of four stages, shown in Figure 2:

1) SubBytes: The SubBytes transformation is a non-linear byte substitution that operates on each byte of the *State* using a substitution table.
2) ShiftRows: In the ShiftRows transformation, the bytes in the last three rows of the *State* are cyclically shifted over different numbers of bytes.
3) MixColumns: Mixing operation which operates on the columns of the *State* using a linear transformation.
4) AddRoundKey: A Round Key is added to the *State* by a simple bitwise XOR operation.

Besides round function, AES algorithm also performs a Key Expansion routine to generate a key schedule. More details about AES can be found in [1].

1-4244-1132-7/07/$25.00 © 2007 IEEE

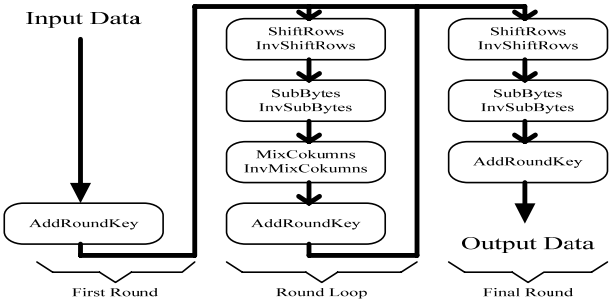

Fig. 1. Data flow.

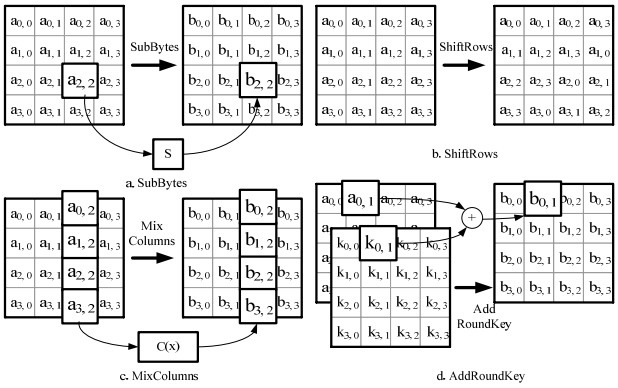

Fig. 2. Operations in AES.

3. Mixed Bus Width Architecture

A. Architecture

In this paper, we propose a new hardware design of AES. The architecture is shown in Figure 3. It refers Satoh's work in [2], Feldhofer's work in [4] and Canright's work in [3]. However, this architecture is totally different from all of them. Satoh's design uses 32-bit bus width, and used 4 S-Boxes to complete 32-bit SubBytes operation. The S-Box and MixColumns module are serially connected in datapath. Feldhofer's design uses 8-bit bus width, one S-Box and 1/4 MixColumns module. The S-Box and MixColumns module are parallel in data path.

The architecture proposed in this paper is a mixed bus width architecture, which is shown in Figure 3. As discussed in Section 2, AES encryption includes four stages: SubBytes, ShiftRows, MixColumns and AddRoundKey. For decryption, the corresponding stages are: InvSubBytes, InvShiftRows, InvMixColumns and AddRoundKey. In hardware implementation, these function modules are shown in Figure 3 (S-Box can support both SubBytes and InvSubBytes). The key points of this architecture include:

a) Mixed bus width. There are two data buses in this architecture. One is 8-bit data bus for SubBytes, the other one is 32-bit data bus for MixColumns.
b) One S-Box: As S-Box costs a lot of hardware resource, there is only one S-Box in this architecture. The design of S-Box is referred from Canright's work in [3] which is the most compact design until now. Satoh's design includes 4 S-Boxes. In this way, the hardware cost of his design is more than our design.
c) 32-bit Mixcolumns: The operation of *MixColumns/InvMixColumns* is 32-bit, so the most efficient bus width for Mixcolums is 32-bit. Feldhofer proposed an 8-bit solution to do this operation. However, 3 additional 8-bit register banks and extra 28 clock cycles are needed. Our 32-bit solution only need 1 clock cycle without adding registers.
d) Parallel architecture in data path: The MixColumns module and S-Box module can execute in parallel. This parallel design can accelerate the speed, and shorten the critical path.

The advantage of mixed bus width architecture is obvious. 32-bit architecture can not reduce the number of S-Box from 4 to 1, so the hardware cost is high. 8-bit architecture needs a large number of extra clock cycles, so the speed is low. Mixed bus width design can both reduce S-Box and support 32-bit MixColumn. It is much more efficient than other two architectures.

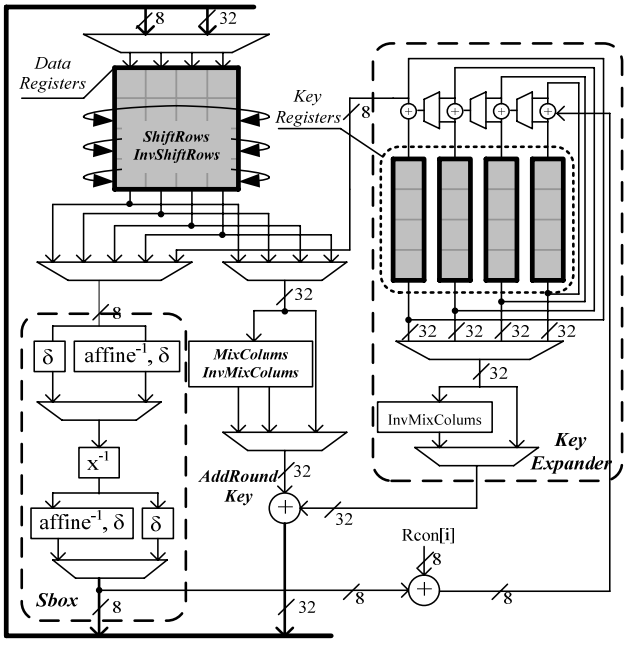

Fig. 3. Mixed bus width architecture for AES.

B. Data Path Configurations and Dataflow

The mixed bus width architecture is much more flexible than other's design. There is more than one kind of data path configurations in this architecture. Each data path configuration can support different operations in parallel. As shown in Figure 4, it has 5 data path configurations:

Config. a (Figure 4.a): This configuration can support SubBytes and ShiftRows. S-Box module is used in this configuration. Since ShiftRows processes last three rows of *States*, the ShiftRows and 1st Row's SubBytes can be done simultaneously. Only 8-bit bus is used in this configuration.

Config. b (Figure 4.b): This configuration can support MixColumns and AddRoundKey. MixColumns module is used in this configuration. 32-bit bus is used here.

Config. c (Figure 4.c): This configuration can support SubBytes, MixColumns and AddRoundKey. Both of S-Box and MixColumns module are used. Both of 8-bit bus and 32-bit bus are used.

Config. d (Figure 4.d): This Configuration can support Mixcolumns, AddRoundKey and KeyExpansion. S-Box, MixColumns and Schedule module are used. Both of 8-bit bus and 32-bit bus are used.

Config. e (Figure 4.e): This Configuration only supports KeyExpansion. S-Box and Schedule module are used.

All of these data path configurations are used in AES encryption/decryption process. By using all of these configurations, some operations in AES can be processed in parallel. The data flow of encryption/decryption is shown in Figure 5. It includes 3 parts:

First Round: As shown in Figure 1, the operation in the first round of AES is AddRoundKey. Here we use data path *Config. a* to do it.

Round Loop*:* All of the operations in AES are used in this step. Some operations can be done in parallel, such as {SubBytes & ShiftRows} (With data path Config. a), {SubBytes & MixColumns & AddRoundKey} (Config. c) and {KeyExpansion & MixColumns & AddRoundKey } (Config. d).

Final Round*:* MixColumns and KeyExpansion are excluded in this step. There are three data path configurations used: Config. a to support SubBytes and ShiftRows, Config. b to support AddRoundkey and Config. c to support SubBytes and AddRoundkey.

By using the data path configurations and data flows described above, a lot of operations can be processed in parallel, so it can save a lot of clock cycles and make this architecture be very efficient.

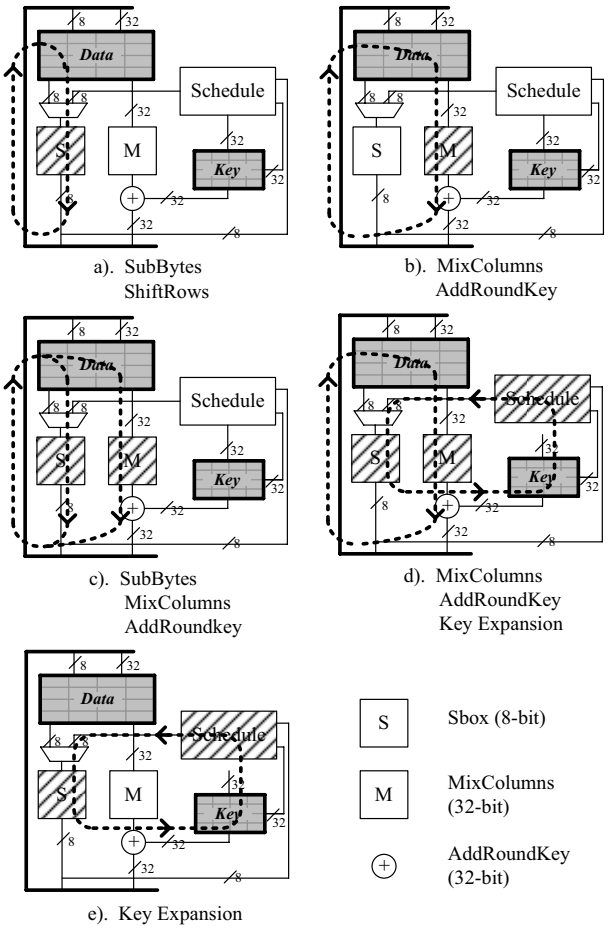

Fig. 4. Data path configurations

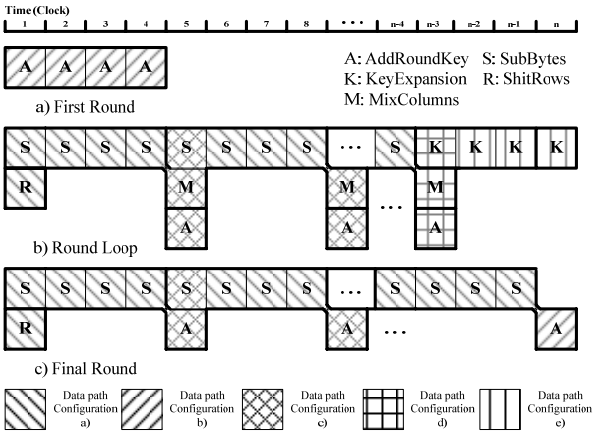

Fig. 5. Data flow and the corresponding data path configurations

Table 1. Clock Cycles

	Number of Cycles
First Round	4
Round Loop	20
Final Round	17
Total (AES 128-bit Encryption)	201 (4+20*9+17)

Table 2. Hardware Cost @ 80MHz, TSMC 0.18um

Components	Gates	%
ShiftRows + Data Registers	1386	29.6%
S-Box	358	7.7%
MixColumns/InvMixcolumns	376	8.0%
Key Expander + Key Registers	1935	41.4%
Controller	247	5.3%
Others	376	8.0%
Total	4678	100%

Table 3. Comparison with other's work

Ref	Tech	Gates	Freq.	Throughput	Year
[2]	0.11um	5398	131MHz	311 Mbps	2001
[4]	0.35um	3595	100KHz	12.6Kbps	2004
[6]	0.25um	12000	100MHz	256Mbps	2006
[7]	0.18um	15073	104MHz	1330Mbps	2006
Ours	0.18um	4678	80MHz	51 Mbps	2007

4. Experimental Results

Table 1 shows the number of clock cycles consumed in each round and in total AES encryption based on proposed architecture.

In order to further reduce hardware cost, we constrained the design by loose frequency to get the lowest hardware cost in synthesis tool. The synthesis results are shown in Table 2. Here we use TSMC 0.18 um standard cell library, and use Synopsys Design Compiler to do synthesis.

Since most of hardware is used for storage in low cost design of AES algorithm (Data registers and Key registers), the throughput-per-gate of low cost design is not as good as high throughput design. The hardware cost is much more important for low cost design.

Table 3 shows the comparison of our design with others'. Our design saves about 13% hardware cost compare to Satoh's design. Even Feldhofer's design has the lowest hardware cost, the througput is too slow to be widely used.

Our design achives both low hardware cost and enough throughput for most of the applications. Especially for mobile system, the power consumption and low hardware cost is the mainly considered point, other than hundreds or thousands of Mbps throughput. Furthermore, our design has shorter critical path than designs which use serial data path. Power consumption can be further reduced while using low voltage power supply.

5. Conclusion

This paper introduces a very low cost implementation of AES algorithm. By using mixed bus width architecture, it can support both 8-bit operation (SubBytes) and 32-bit operation (MixColumns, AddRoundKey), so the number of S-Box can be reduced to 1 to save hardware cost. This design is very suitable to be used in low power and low hardware cost systems.

Acknowledgments

This research is supported by CREST, JST.

References

[1] National Institute of Standards and Technology (U.S.). Advanced Encryption Standards (AES). FIPS Publication 197, (2001).

[2] A. Satoh, S. Morioka, K. Takano, S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, pp.239 – 254 (2001).

[3] D. Canright, "A Very Compact S-Box for AES," Cryptographic Hardware and Embedded Systems – CHES, September, pp.441 – 455 (2005).

[4] M. Feldhofer, S. Dominikus, J. Wolkerstorfer, " Strong Authentication for RFID Systems Using the AES Algorithm," Cryptographic Hardware and Embedded Systems - CHES 2004, Volume 3156, pp.357-370 (2004).

[5] S. Morioka, A. Satoh, "An Optimization S-Box Circuit Architecture for Low Power AES Design", CHES 2002, LNCS 2523, pp. 172-186 (2002).

[6] Jia Zhao, Xiaoyang Zeng, Jun Han, Jun Chen, "Very Low-cost VLSI Implementation of AES Algorithm", IEEE Asian Solid-State Circuits Conference, pp. 223 - 226 (2006).

[7] Shen-Fu Hsiao, Ming-Chih Chen, Chia-Shin Tu, "Memory-free low-cost designs of advanced encryption standard using common subexpression elimination for subfunctions in transformations", IEEE Transactions on Circuits and Systems I, Volume 53, Issue 3, March, pp.615 – 626 (2006).