A 3.84 Gbits/s AES Crypto Coprocessor with Modes of Operation in a 0.18-µm CMOS Technology

Alireza Hodjat[1], David D. Hwang[1], Bocheng Lai[1], Kris Tiri[1], Ingrid Verbauwhede[1,2]

[1] University of California, Los Angeles

[2] Katholieke Universiteit Leuven

{ahodjat, dhwang, bclai, tiri, ingrid} @ ee.ucla.edu

ABSTRACT

In this paper an AES crypto coprocessor that is fabricated using a 0.18-µm CMOS technology is presented. This crypto coprocessor performs the AES-128 encryption in both feedback and non-feedback modes of operation. A maximum throughput of 3.84 Gbits/s is achieved at a 330 MHz clock frequency for ECB, OFB, and CBC modes of operation. This crypto coprocessor can be programmed using the memory-mapped interface of an embedded CPU core and is tested using a LEON 32-bit (SPARC V8) processor in the ThumbPod secure system-on-chip.

Categories and Subject Descriptors

B.7.1 [**Hardware**]: Integrated Circuits – *Algorithms implemented in hardware*

General Terms

Design, Security.

Keywords

Advanced Encryption Standard (AES), Cryptography, Crypto-processor, Security, Hardware architectures, ASIC, FPGA, VLSI.

1. INTRODUCTION

High throughput crypto coprocessors are needed for acceleration of encryption functions in different types of systems. As the demand for secure communications increases, high-throughput encryption on both wired and wireless networks is growing more necessary. In recent years, even embedded devices such as cellular phones and 802.11b-equipped PDAs require fast encryption. As data rates continue to increase on such systems, software-based encryption is inadequate; high-speed, low-area cores are necessary. This paper presents such an AES crypto module with high-throughput that is built as a coprocessor with a memory-mapped interface. As such, it can be used with a large class of embedded cores. In our case, we have implemented it as a coprocessor to the LEON (SPARC V8) embedded processor for a secure embedded authentication system (called ThumbPod).

GLSVLSI'05, April 17–19, 2005, Chicago, Illinois, USA.

The Advanced Encryption Standard [1] is the most recent symmetric-key algorithm standard used in different security protocols. An encryption algorithm is never used stand-alone for security reasons. Therefore, it is combined with so-called modes of operation. Reference [2] shows the US National Institute of Standards and Technology recommendations for block cipher modes of operation. These are Electronic Code Book (ECB), Output Feed-Back (OFB), Cipher Block Chaining (CBC), and Cipher Feed-Back (CFB) modes of operation. Due to the feedback line in the OFB and CBC modes, the encryption algorithm can not be pipelined. Therefore, achieving a throughput of multiple Gbits/s encryption is a challenge when the AES algorithm is used in the above modes of operations. This paper presents a fabricated AES crypto coprocessor chip that can achieve a maximum throughput of 3.84 Gbits/s for all of the above feedback and non-feedback modes using a 0.18-µm CMOS technology.

The rest of this paper is organized as follows: Section 2 provides the related work. In section 3 the AES algorithm is presented. Section 4 presents the architecture of our crypto coprocessor chip. Section 5 describes the crypto coprocessor interface. Section 6 presents the test results, and the comparison with other published results is in section 7. The conclusion is given in section 8.

2. RELATED WORK

The Advanced Encryption Standard was accepted as a FIPS standard in November 2001 [1]. References [3, 4 and 5] are some of the early implementations of the Rijndael algorithm before it was accepted as the Advanced Encryption Standard. Reference [6] is the first ASIC implementation of Rijndael on silicon. Other ASIC implementations are [7, 8, and 9]. These references mainly focus on area efficient implementations of the AES algorithm using Sbox (byte substitution phase) optimizations. They all use the suggestion of Rijmen [12] that proposed a way of optimization of the Sboxes based on transforming the original field of $GF(2^8)$ to a composite field of $GF((2^4)^2)$ or $GF(((2^2)^2)^2)$. These all focus on one round implementation of the AES algorithm without pipelining, and they can only provide a throughput rate of between 2 to 3 Gbits/s. Reference [13] can achieve the throughput of 10 Gbps (synthesis result using a 0.13µm CMOS library) by implementing binary decision diagram (BDD) circuit architecture and TBoxes which are the combination of Sboxes and the mix-column phase of the AES algorithm. There are several implementations for FPGA that can achieve a throughput rate of 1 to 20 Gbits/s because they unroll the encryption rounds and use pipelining. However, due to the use of pipelining they are not suitable for feedback modes of operation like OFB and CBC. Some of these references are [14, 15, 16, 17, 18, and 19]. Also

[20] presents pipeline architectures that can achieve more than 30 Gbits/s throughput for ASIC platforms. This paper presents a non-pipelined AES coprocessor that is based on the one round implementation of the encryption algorithm with an on-the-fly key scheduling unit with minimum critical path delay. This coprocessor can achieve a maximum throughput of 3.84 Gbits/s for both feedback and non-feedback modes of operation, which is the fastest published AES encryption rate on silicon.

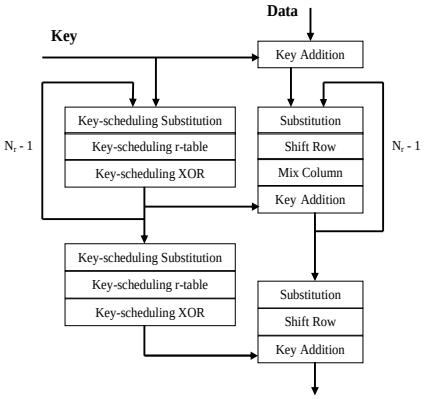

Figure 1: Advanced Encryption Standard

3. ADVANCED ENCRYPTION STANDARD

Figure 1 shows the different steps of the AES algorithm [1]. The AES algorithm is performed in N_r number of rounds. The architecture of one round contains two different datapaths, the encryption datapath and the key scheduling datapath. In the AES algorithm, the data block is 128 bits long and the key size can be 128, 192, or 256 bits. The size of the key defines the number of rounds that the algorithm is repeated. The value N_r is equal to 10, 12, or 14 for the key length of 128, 192, or 256 bits, respectively.

There are different steps in each round of the encryption datapath:

- Byte substitution: This step is a non-linear operation that substitutes each byte of the round data independently according to a substitution table.
- Shift row: This step is a circular shifting of bytes in each row of the round data. The number of shifted bytes is different for each row.
- Mix column: In this step the bytes of each column are mixed together. This is done by multiplying the round data with a fixed polynomial modulo $x^4 + 1$.
- Add key: In this step the round data is XORed with the round key.

All the above four steps are required for every round except that the last round does not include the mix column phase. Similar steps are followed in the key scheduling flow.

4. COPROCESSOR ARCHITECTURE

4.1 Crypto Coprocessor Block Diagram

Figure 2 shows the block diagram of the different units of the proposed AES-based crypto coprocessor. The AES core consists of the encryption datapath and the key scheduling datapath. There is logic around the AES core to implement different modes of operations. The input and output interfaces take care of reading input data and writing encrypted output. The crypto controller reads in the instructions and controls other units.

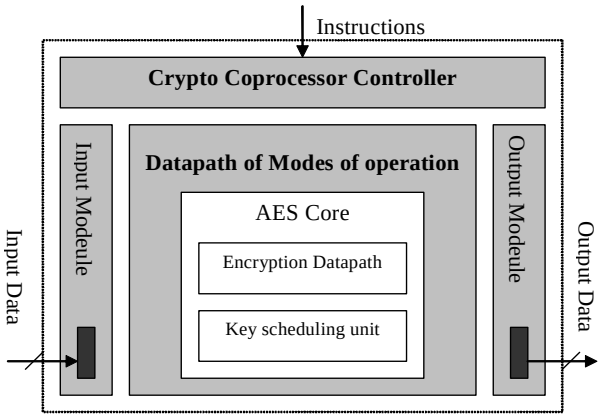

Figure 2: Block Diagram of the Crypto Coprocessor

4.2 Datapath of Modes of Operation

Different feedback and non-feedback modes of operation are required for the secure encryption of data. In our application ECB, OFB, and CBC modes of operations are used for data confidentiality and authentication. Figure 3 shows how these modes are implemented for a typical block cipher. Due to the feedback in these modes of operation the block cipher can not be pipelined. The next section presents the details of the non-pipelined AES core that is designed for this crypto coprocessor. Figure 4 shows the datapath of modes of operation for the AES core. Different registers that contain intermediate values of data and key are used as well as logic that implements the ECB, OFB, and CBC modes of operations.

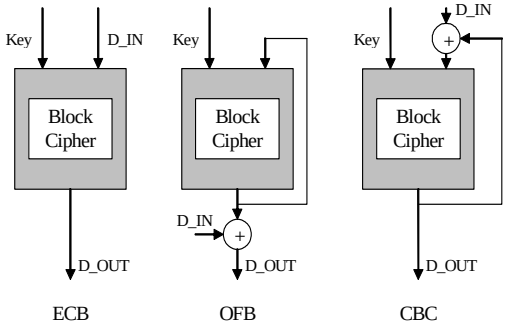

Figure 3: Feedback and non-feedback modes of operations

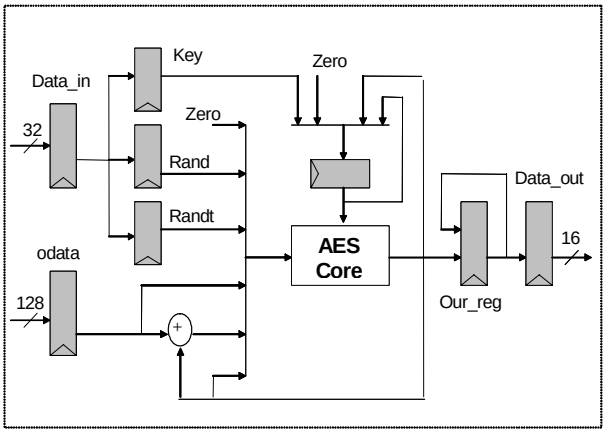

Figure 4: Datapath of modes of operation

4.3 Non-pipeline AES-128 Core

4.3.1 Encryption Datapath

The encryption datapath of the non-pipelined AES-128 core is shown in Figure 5. This datapath is based on the single round implementation of the AES algorithm. The same datapath is used for the 10 rounds in the AES-128 algorithm. Each round is performed in a single clock cycle. Except the first round (round zero) that only performs the key addition phase, all other 10 rounds perform the four different steps of the AES algorithm: Byte substitution, Shift row, Mix column (not done last round), and Key addition. Therefore, it takes total of 11 cycles to encrypt a 128-bit block of data.

The AES-128 encryption datapath is optimized for the goal of minimum delay for each round. The look-up table implementation of the byte substitution phase, which is the fastest implementation of this phase of the AES algorithm [6], is used. The mix column step is implemented using a chain of XORs which results in the minimum delay implementation for this unit. Overall, the total combinational delay of all the four steps of the AES-128 are minimized so that each round of the AES-128 algorithm is performed in a single clock cycle at a maximum clock frequency.

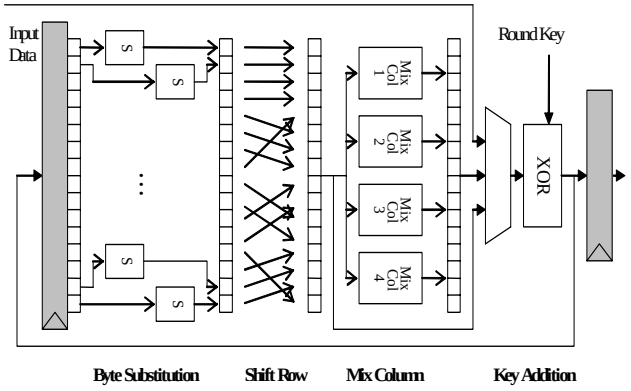

Figure 5: Details of the encryption datapath

4.3.2 Key scheduling Datapath

An on the fly key scheduling datapath (figure 6) that performs the similar steps to the encryption datapath is designed for the AES core. This is due to the fact that in the AES-128 algorithm the key size is fixed to 128 bits long and therefore, the critical path of the encryption datapath and the key scheduling datapath are balanced.

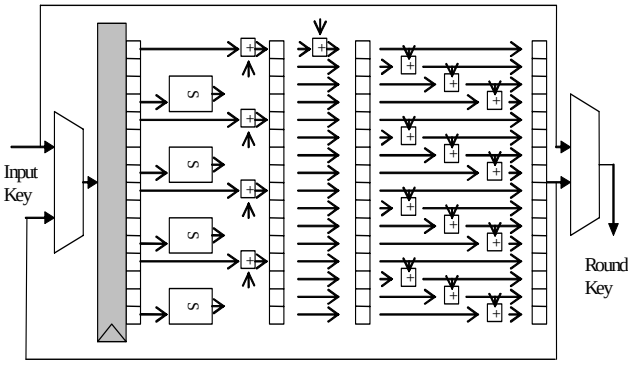

Figure 6: Details of the key scheduling unit

5. CRYPTO COPROCESSOR INTERFACE

The proposed crypto coprocessor can be programmed through the memory-mapped interface of an embedded CPU core. Three registers connect the embedded CPU to the IO ports of the crypto coprocessor which are the *Instruction* port, *Input_data* port, and *Output_data* port (Figure 2). The embedded CPU core can read or write to these registers by accessing different memory locations. The memory-mapped interface decodes the memory addresses and updates the registers' values. In our case, the crypto coprocessor is attached to the memory-mapped interface of the LEON (SPARC V8) embedded processor.

To provide context to this work, the cryptographic engine described in this paper is one subsection of a fabricated coprocessor IC. The fabricated coprocessor IC is part of a ThumbPod embedded system, which is a portable biometric and cryptographic device. In this application, the coprocessor is used to perform a variety of embedded biometric and cryptographic protocols for secure wireless authentication. The IC consists of a number of components such as the cryptographic engine (this paper), a biometric matching algorithm processor, and a memory component for storage of biometric data. The chip monograph of the AES crypto coprocessor is shown in figure 7.

6. TEST RESULTS

The measurements results of our chip test set-up is shown in Table 1. The cryptographic engine was able to operate in feedback mode at 330 MHz, which equates to 3.84 Gb/s. Since one block of 128-bit data is encrypted every 11 cycles, therefore the throughput is calculated by multiplying the clock frequency times 128 divided by 11. The core area of the cryptographic engine was 0.79 mm^2, roughly 40% of the total core area of the fabricated IC. The cryptographic engine could inter-operated with the other subsystems at a maximum frequency of 288 MHz. During self-test at 50 MHz, the coprocessor IC consumed 54 mW.

Table 1. Measurement results of the AES crypto coprocessor

	AES Coprocessor
Core Area	0.79 mm^2
Maximum Clock Frequency	330.0 MHz
Power Consumption @ 1.8V, 50 MHz	54 mW
Maximum Throughput	3.84 Gbits/s

7. RESULTS COMPARISON

Table 2 compares the performance results of our AES crypto coprocessor chip with other working silicon AES chips. Other high speed AES chips that are reported in Table 2 can achieve a throughput rate of between 2 to 3 Gbits/s. Reference [7] has the maximum throughput among these references, however their implementation is a pipelined version of the AES algorithm and can not be used in the OFB and CBC modes of operation. On the other hand reference [11] is a nice balanced AES crypto chip that can perform both feedback and non-feedback modes of operations at the maximum throughput of 2.1 Gbits/s. As shown in Table 3, other AES chips sit in between these two references in terms of maximum throughput and supported modes of operation. On the

other hand, our chip can achieve the maximum throughput of 3.84 Gbits/s for all the supported modes of operation: EBC, OFB, and CBC. Moreover, the modes of operation are already implemented in hardware inside of our chip and there is no overhead in programming and software when the modes of operation are used.

8. CONCLUSION

This paper presented a 3.84 Gbits/s non-pipelined AES encryption coprocessor. The proposed crypto coprocessor is working silicon that is fabricated using a 0.18-µm CMOS technology. It supports the feedback and non-feedback modes of operation: ECB, OFB, and CBC. The maximum throughput of 3.84 Gbits/s is achieved at 330 MHz clock frequency for all of the supported modes. This crypto coprocessor is built such that it can be connected to many embedded processors, such as ARM/LEON/POWERPC, through the memory mapped interface.

9. ACKNOWLEDGMENT

The authors acknowledge supports of the NSF (CCR-0098361), Sun, UC Micro, Panasonic, Atmel, the Fannie and John Hertz Foundation (DH), Space and Naval Warfare Systems Center - under contract No. N66001-02-1-8938, and SRC (2003-HJ-1116).

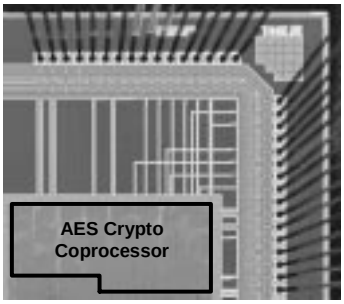

Figure 7: Chip micrograph of the AES crypto coprocessor

Table 2. Comparison with other implementations

	This paper	Su [7]	Satoh [8]	Kim [10]	Verbauwhede [6]	Gurkaynak [11]
Throughput	3.84 Gbits/s	2.97 Gbits/s	2.6 Gbits/s	2.33 Gbits/s	2.29 Gbits/s	2.12 Gbits/s
Clock Frequency	330.0 MHz	250 MHz	224.2 MHz	465 MHz	154 MHz	166 MHz
Core Area	0.79 mm^2	1.62 mm^2	0.205 mm^2	N/A	3.96 mm^2	3.56 mm^2
Technology	0.18 µm	0.25 µm	0.11 µm	0.18 µm	0.18 µm	0.25 µm
Supported modes	ECB, OFB, CBC	ECB only	ECB	ECB	ECB	ECB, OFB, CFB. CBC

10. REFERENCES

[1] National Institute of Standards and Technology (U.S.), Advanced Encryption Standard.

[2] M. Dworkin, SP 800-38A 2001, "Recommendation for Block Cipher Modes of Operations," December 2001.

[3] T. Ichikawa et al, "Hardware Evaluation of the AES Finalists,"Proc. 3th AES candidate conference, April 2000.

[4] K. Gaj and P. Chodowiec, "Comparison of the Hardware Performance of the AES Candidates using Reconfigurable Hardware",Proc.3th AES candidate conference, April 2000.

[5] V. Fischer, "Realization of the Round 2 Candidates using Altera FPGA", Proc.3th AES candidate conf., April 2000.

[6] I. Verbauwhede, P. Schaumont, H. Kuo, "Design and Performance testing of a 2.29 Gb/s Rijndael Processor", IEEE Journal of Solid-State Circuits, Pages:569–572, 2003.

[7] C. Su, T. Lin, C. Huang, and C. Wu, "A High-Throughput Low-cost AES processor," IEEE Communication Magazine, Vol. 41, Issue 12, pp. 86-91, December 2003.

[8] A. Satoh, S. Morioka, K. Takano, S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," in Proc.ASIACRYPT 2001, LNCS2248, pp.239-254, 2001.

[9] J. Wolkerstorfer, E. Oswald, M. Lamberger, "An ASIC Implementation of the AES Sboxes," in Proc. RSA Conference 2002, San Jose, CA, February 2002.

[10] N.S. Kim, T. Mudge, R, Brown, "A 2.3 Gb/s Fully Integrated and Synthesizable AES Rijndael Core," in Proc. IEEE Custom Integrated Circuits Conference, pp. 193-196, September 2003.

[11] F. Gurkaynak, D. Hug, and H. Kaeslin, "A 2 Gb/s Balanced AES Crypto-Chip Implementation", GLSVLSI 2004.

[12] V. Rijmen, "Efficient Implementation of the Rijndael S-box", http://esat.kuleuven.ac.be/~rijmen/rijndael/sbox.pdf

[13] S. Morioka, A. Satoh, "A 10-Gbps Full-AES Design with a Twisted BDD S-Box Architecture", IEEE Trans. On VLSI, Vol. 12, No. 7, July 2004.

[14] M. McLoone, J. McCanny, "High Performance Single Chip FPGA Rijndael Algorithm Implementations"CHES 2001.

[15] A. Elbirt, et al, "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," IEEE Trans. of VLSI Systems, pp. 545-557, August 2001.

[16] Standaert et al, "Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware: Improvements and Design Tradeoffs," CHES 2003, LNCS 2779, pp. 334-350.

[17] Saggese et al, "An FPGA-Based Performance Analysis of the Unrolling, Tiling, and Pipelining of the AES Algorithm," FPL 2003, LNCS 2778, pp. 292-302, 2003.

[18] Jarvinen et al, "A Fully Pipelined Memoryless 17.8 Gbps AES-128 encryptor", International Symposium on Field Programmable Gate Arrays, pp. 207-215. 2003.

[19] Alireza Hodjat, Ingrid_verbauwhede, "A 21.54 Gbits/s fully pipelined AES processor on FPGA", IEEE Symposium on Field -Programmable Custom Computing Machines, April 2004.

[20] Alireza Hodjat, Ingrid Verbauwhede, "Minimum Area Cost for a 30 to 70 Gbits/s AES Processor", Proceedings of IEEE computer Society Annual Symposium on VLSI, Pages: 83-88, February 2004.