A 2.29 Gbits/sec, 56 mW Non-Pipelined Rijndael AES Encryption IC in a 1.8V, 0.18 μm CMOS Technology

Henry Kuo, Ingrid Verbauwhede, Patrick Schaumont
(henrykuo@ee.ucla.edu, Ingrid@ee.ucla.edu, schaum@ee.ucla.edu)

Electrical Engineering Department, University of California Los Angeles, Los Angeles, CA.

Abstract

In October 2000 the National Institute of Standard and Technology (NIST) chose the Rijndael algorithm as the new Advanced Encryption Standard (AES). In this paper we present an ASIC implementation of the Rijndael core. The core includes a non-pipelined encryption datapath with an on-the-fly key schedule data path. At a nominal 1.8V, the IC runs at 125 MHz resulting in a throughput of 2.29 Gbits/sec while consuming 56 mW. At 1.95V, the chip can operate up to 154 MHz with an equivalent throughput of 2.8 Gbits/sec and consumes 82 mW.

Introduction

According to our knowledge, this contribution describes the *first* ASIC encryption core that implements the Rijndael AES encryption standard. At a clock frequency of 125 MHz and 1.8V, it realizes a throughput of 2.29 Gbits/sec at 56 mW. The widespread adoption of distributed, wireless, and mobile computing makes the inclusion of privacy, authentication and security in general a necessity. In October 2000, the NIST adopted the Rijndael algorithm as the new AES [1]. It will gradually replace DES and triple DES for encryption purposes in commercial applications, ranging from Internet applications such as IPSEC to wireless LAN applications such as the IEEE 802.11 WLAN standards.

System Architecture

The architecture of the encryption IC is shown in Fig. 1. There are four main data path modules: the encryption core, the key scheduling core, an input module and an output module. There is a main processor FSM that accepts the instructions listed in Table 1, and decodes the instructions for the local FSMs of the data path modules. The encryption data path is shown on Fig. 2. It consists of 4 main steps: a substitution step, a shift row step, a mix column step and a subkey addition step.

The substitution step consists of Sboxes. The shift row step consists of a cyclic-shifting of the bytes within the rows. The block diagram for the mix column is shown in Figure 3. Only byte 0 of the calculation is shown. The key addition is straight forward XOR operations between the data and the key.

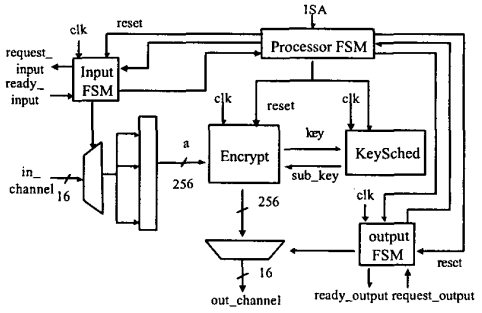

Fig. 1 Overall chip architecture of the Rijndael IC.

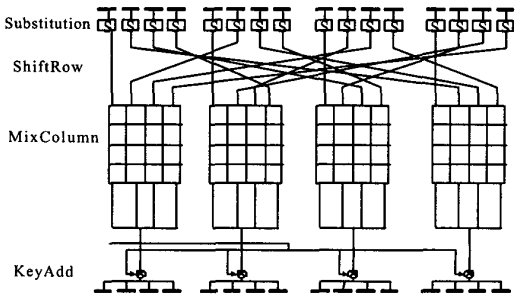

Fig. 2 Architecture of Encryption Datapath.

The key-scheduling module is shown in Fig. 4 and 5. The chip contains hardware to generate one set of sub-key in one clock cycle, and the same hardware is used to generate each subsequent keys parallel to the encryption module. We designed a separate key selection module to allow flexibility in key size as well.

0-7803-7250-6/02/$10.00 © 2002 IEEE

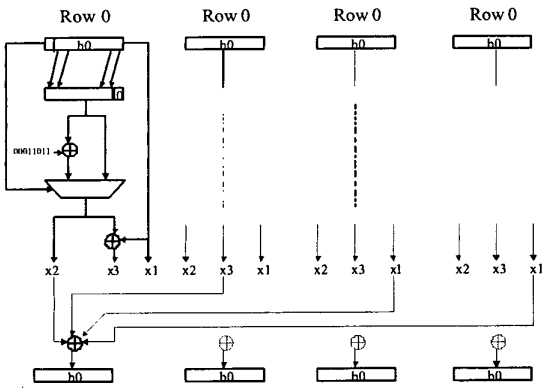

Fig.3. Block diagram for Mix Column.

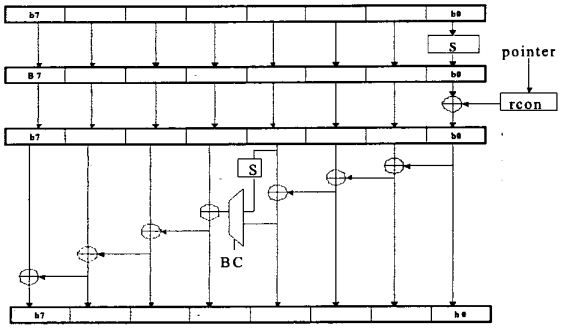

Fig. 4 Block diagram for Key Scheduling.

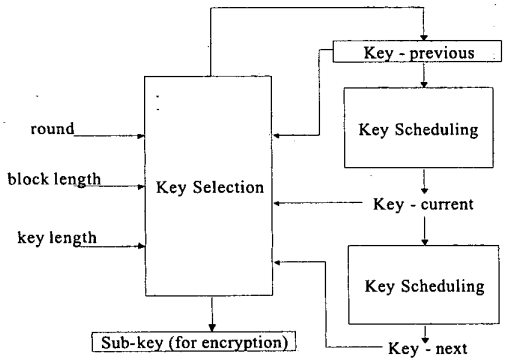

Fig. 5 Block diagram for the added Key Selection module.

A. Feedback Modes

A few key architecture decisions have been made, to make sure that this implementation can be used in combination with the standard modes of operation [2]. Indeed, the cipher block chaining (CBC) mode, cipher feedback (CFB) and output feedback (OFB) modes, are considered more secure than the straightforward electronic code book (ECB) mode. In each of the three first modes, the output of the AES algorithm is fed back to the input. Therefore, it prohibits a pipelined implementation of the AES algorithm. Only the ECB mode can be pipelined, but it is not recommended for usage [2]. Thus we have chosen for a maximum parallel implementation of one "round" (i.e. iteration step) of the AES algorithm. This architecture implements *one round of the AES algorithm in one clock cycle*. This means that the S-boxes (the largest component in the algorithm) are repeated 32 times. Sub keys are generated "on the fly" in this architecture, which eliminates the need to store the sub keys.

Reset		0000
Set Key Length	128	0010
	192	0011
	256	0100
Set Block Length	128	1010
	192	1011
	256	1100
Input Key		0001
Input Data		1001
Encryption		1101
Decryption		0101
Encryption – Feedback Mode (for testing)		1110
Decryption – Feedback Mode (for testing)		0110
Output Data		0111

Table 1. Instruction Set.

B. I/O Implementation

The Input and Output module are asynchronous modules. The input is 16-bit wide and data is placed in an interface register by blocks of 2 bytes. Similar, the cipher text is placed in an output register. The output FSM will place blocks of 16 bits on the output pins. This asynchronous handshaking allows us to test the encryption core at a high clock frequency, without the I/O pads or the input/output interface being in the critical path.

C. Optional Changes

The IC described in this contribution implements the original Rijndael algorithm as it was submitted to NIST [3]. This means that any of the nine combinations of a data length of 128, 192 and 256 bits and a key length of 128, 192 and 256 bits can be implemented. This requires a flexible key scheduling module of Figure 5, where the key scheduling hardware is implemented twice, combined with a sub key selection module, based on the round, the key

length and the block length. Hence, one sub key generation can be generated in one clock cycle. However, the NIST has adopted a simplified version that only implements a data width of 128 bits and a programmable key length of 128, 192 or 256 bits. This means that our design can be greatly simplified, especially the key scheduling module of Figure 3. Also, for this first version, we opt not to implement the decryption mode yet.

Test Results

The design was implemented in a 0.18μm 1.8V CMOS standard cell library of National Semiconductor. The design was first described in Verilog and synthesized with Synopsys design compiler. Place and Route are performed with Avanti tools. 14 out of 16 samples are fully functional and the Schmoo plot with the minimum clock periods are shown on Fig. 6. Fig. 7 shows the power consumption as a function of the clock period. At the nominal voltage of 1.8V, the clock frequency of 125 MHz, power consumption is 54mW, at 1.95V the clock frequency is 154 MHz and 82mW. The die photo is shown in Fig. 8.

Feature	Value			
Technology	0.18 μm CMOS			
Core voltage	1.8 V			
I/O ring	3.3 V			
Package	68 pin plastic PLCC			
Clock period	6.5 ns at 1.95V, 7.5 ns at 1.9V, 8 ns at 1.8V			
Throughput (Gbits/s at 1.8 V)	Data \ Key	128	192	256
	128	1.6	1.33	1.14
	192	2.0	2.0	1.71
	256	2.29	2.29	2.29
Power (core)	54 mW at 1.8 V, 67 mW at 1.9V, 82mW at 1.95V			
Gate count	Encryption	Key-Scheduling	Total	
	99,300	66,100	173,000	

Table 2. Main chip characteristics.

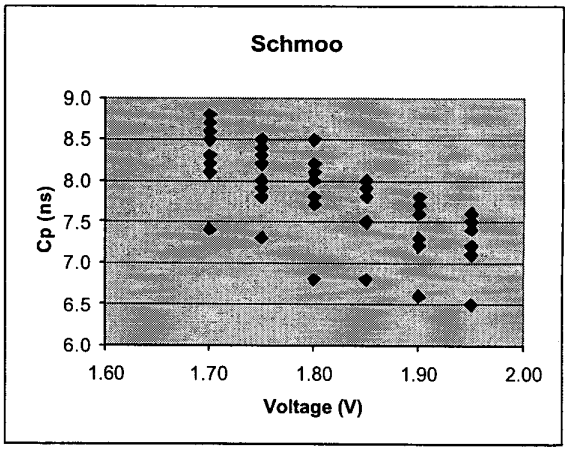

Fig. 6. Schmoo plot.

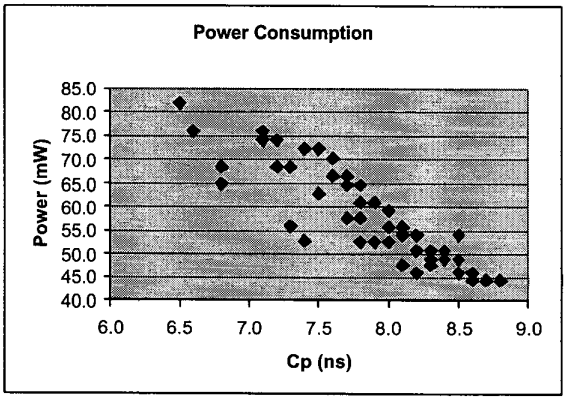

Fig. 7. Power consumption as a function of clock period.

Results Comparisons

There have been a few other AES implementations in software and FPGA. The FPGA design reported in [4] is a pipelined implementation that runs up to 6.95Gbit/sec for a 128-bit data/128 bit key. However this is a pipelined implementation and thus its throughput has to be divided by the number of pipeline stages (at least 10) when it is used in combination with a feedback mode of operation. A different FPGA implementation is described in [8]. It describes both a pipelined and non-pipelined implementation, but only of the encryption core. It does not include decryption nor key scheduling. The best pipelined implementation reaches 1.94 Gbit/s, the best feedback mode implementation reaches 300 Mbit/s.

There have been simulations of standard cell implementations reported in [5] and [6]. The first simulation describes a fully loop unrolled implementation (without pipelining between the stages). It uses 613 Kgates and runs at 1.95 Gbits/sec in a 0.35 μm CMOS technology. This

implementation is only an encryption module for a 128-bit encryption data path with a 128 bit key and it does not include the key scheduling nor the decryption module. In [6] simulations are made for a 128 bit data, and 128, 192 and 256 bit key implementation. The non-pipelined version simulations result at maximum 450 Mbits/sec in a 0.5 μm CMOS standard cell library. On a 933 MHz Pentium processor, the fastest software implementations run at 325 Mbits/sec for a 128-bit key, 275 Mbits/sec for 192 bit key and 236 Mbits/sec for a 256-bit key [7]. Data is 128 bit and software loop unrolling (the software equivalent of pipelining) and other optimizations are performed.

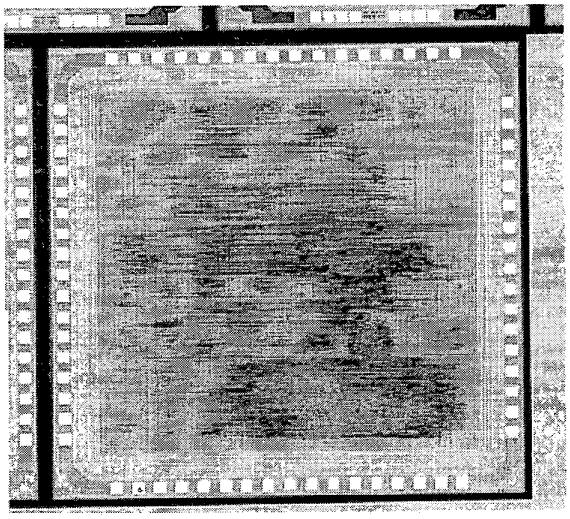

Fig. 8. Die photo of Rijndael AES IC.

Future Developments

The current critical path, 10ns, sits in the key-scheduling module, while the encryption core has a critical path of 6ns. Simplifying the logic to only support the 128 bit data version, will reduce the logic of the encryption module by half but will not effect that much the critical path. But the key schedule module will simplify even more and we expect that it will simplify the decision logic. Thus we expect a simplified version of this IC to run at least at 200 MHz in less than 100,000 gates. This will make the 128-bit data version run above 2.5Gbits/s at nominal voltage.

Conslusion

In this paper we described the first known ASIC implementation of the AES Rijndael algorithm. The chip can accept data and keys of 128, 192, and 256 bits. Using 0.18μm CMOS technology to fabricate the ASIC, we achieve a maximum throughput of 2.29Gbits/sec at 1.8V. At this voltage the chip dissipates around 56mW of power. If we increase the voltage to 1.95V, which is the highest we could get, the encryption core achieves a throughput of 2.8Gbits/sec with a power consumption of 82mW. Our results are the fastest non-pipelined implementation so far, and are even faster than most pipelined version. We also found out that by reducing the key-scheduling datapath, the chip is expected to run at over 2.5Gbits/sec at 1.8V.

References

[1] "Draft FIPS for the AES", available from http://csrc.nist.gov/encryption/aes, February 2001.

[2] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[3] J. Daemen and V. Rijmen, "The block cipher Rijndael" in *Smart Card Research and Applications*, Lecture Notes in Computer Science LNCS 1820, J.-J. Quisquater, B. Schneier, Eds, Springer Verlag, 2000, pg. 288-296.

[4] M. McLoone, J. McCanny, "High Performance Single-Chip FPGA Rijndael Algorithm Implementations," Proceedings Cryptographic Hardware and Embedded Systems Workshop, CHES, Paris, May 2001.

[5] T. Ichikawa, T. Kasuya, M. Matsui, "Hardware Evaluation of the AES Finalists," in AES3: the third Advanced Encryption Standard Candidate conference, New-York, April 13-14, 2000.

[6] M. Bean, C. Ficke, T. Rozylowicz, B. Weeks, "Hardware Performance simulations of Round 2 Advanced Encryption Standard Algorithms," available from http://csrc.nist.gov/encryption/aes/round2/NSA-AESfinalreport.pdf.

[7] B. Gladman, "The AES Algorithm (Rijndael) in C and C++, performance of the optimized implementation," from http://fp.gladman.plus.com/cryptography_technology/rijndael/index.htm

[8] A. Elbirt, W. Yip, B. Chetwynd, C. Paar, "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists, IEEE Transactions on VLSI Systems, Vol. 9, no.4, August 2001.