IJMTST | Volume: 2 | Issue: 08 | August 2016 | ISSN: 2455-3778

Novel Adaptive Hold Logic Circuit for the Multiplier using Add Round Key and Parallel AES

Mr. S. Mohan Das[1] | Prem Joshua M[2]

[1]Associate Professor, Department of ECE, SVR Engineering College, Nandyal.
[2]PG Scholar, Department of ECE, SVR Engineering College, Nandyal.

ABSTRACT

Digital multipliers are among the most critical arithmetic functional units in many applications, such as the Fourier transform, discrete cosine transforms, and digital filtering. The through put of these applications depends on multipliers, if the multipliers are too slow, the performance of entire circuits will be reduced. The negative bias temperature instability effect occurs when a PMOS transistor is under negative bias (Vgs = −Vdd), increasing the threshold voltage of a PMOS transistor and reducing the multiplier speed. Similarly, positive bias temperature instability occurs when an NMOS transistor is under positive bias. Both effects degrade the speed of the transistor and in the long term, the system may be fail due to timing violations. Therefore, it is required to design reliable high-performance multipliers. In this paper, we implement an aging aware multiplier design with a novel adaptive hold logic (AHL) circuit. The multiplier is able to provide the higher throughput through the variable latency and can adjust the adaptive hold logic (AHL) circuit to lessen performance degradation that is due to the aging effect. The proposed design can be applied to the column bypass multiplier.

KEYWORDS: *Advanced Encryption Standard, Sub bytes, Shift row, mixed column, Add round Key and Parallel AES*

I. INTRODUCTION

Cryptography is the practice and study of hiding information. Applications of cryptography include ATM cards, computer passwords, until modern times cryptography referred almost exclusively to encryption, which is the process of converting ordinary information into unintelligible gibberish. Decryption is the reverse, in other words, moving from the unintelligible cipher text back to plaintext. A cipher is a pair of algorithm which creates the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a key. With the development of information technology, protecting sensitive information via encryption is becoming more and more important to daily life. In 2001, the National Institute of Standards and Technology selected the Rijndael algorithm as the Advanced Encryption Standard (AES),which replaced the Data Encryption Standard (DES). Since AES has been widely used in a variety of applications, such as secure communication systems, high-performance database servers, digital video/ audio recorders, and smart cards.

The Advanced Encryption Standard (AES) is an encryption standard that comprises three block ciphers, AES-128, AES-192and AES-256, adopted for different applications. AES is one of the most popular algorithms used in symmetric key cryptography. It is available in many different encryption packages. AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware, is relatively easy to implement, and requires little memory. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits. Assuming one byte equals 8 bits, the fixed block size of 128 bits is 16 bytes. AES operates on a 4×4 array of bytes,

termed the state. A set of reverse rounds are applied to transform cipher-text back into the original plain-text using the same encryption key. The proposed architecture is FPGA based architectures with high-speed and low area constraints for suitable implementation of Advanced Encryption Standard (AES). The main focus of this paper is to compare different design architectures existing in literature with the proposed ones, based on application specific constraints. Most AES calculations are done in a special field. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plain-text into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key.

II. Related Work

Verbauwhede, P. Schaumont, and H. Kuo [1] proposed the design and performance testing of an Advanced Encryption Standard (AES) compliant encryption chip that delivers 2.29 GB/s of encryption throughput at 56mW of power consumption in a0.18- m CMOS standard cell technology. This integrated circuit implements the Rijndel encryption algorithm, at any combination of block lengths 128, 192, and 25 bits) and key lengths (128, 192, or 256bits). We present the chip architecture and discuss the design optimizations. We also present measurement results that were obtained from a set of 14 test samples of this chip.[2]proposed a high-speed AES IP-core, which runs at 880 MHz on a 0.13- m CMOS standard cell library, and which achieves over10-Gbps throughput in all encryption modes, including cipher block chaining (CBC) mode. Although the CBC mode is the most widely used and important, achieving such high throughput was difficult because pipelining and/or loop unrolling techniques cannot be applied. To reduce the propagation delays of the S-Box, the slowest function block, we developed special circuit architecture that we call twisted-binary decision diagram (BDD), where the fan out of signals is distributed in the S-Box circuit. Our S-Box is 1.5 to 2 times faster than the conventional S-Box implementations. The T-Box algorithm, which merges the S-Boxand another primitive function into a single function, is also used for an additional speedup. [3] describes the area-through puttrade-off for an ASIC implementation of the Advanced Encryption Standard(AES).Different pipelined implementations of the AES algorithm as well as the design decisions and the area optimizations that lead to a low area and high throughput AES encryption processor are presented. With loop unrolling and outer-round pipelining techniques, throughputs of 30 Gigabits/s to 70Gigabits/s are achievable in a 0.18-_m CMOS technology. Moreover, by pipelining the composite field implementation of the byte substitution phase of the AES algorithm (inner-round pipelining), the area consumption is reduced up to 35 percent. By designing an offline key scheduling unit for the AES processor the area cost is further reduced by 28 percent, which results in a total reduction of 48 percent while the same throughput is maintained. Therefore, the over 30 Gigabits/s, fully pipelined AES processor operating in the counter mode of operation can be used for the encryption of data on optical links.

III. Implementation

3.1 Existing System:

Composite fields are frequently used in implementations of Galois Field arithmetic. In cases where arithmetic operations relyon table lookups, subfield arithmetic is used to reduce lookup-related costs. This technique has been used to obtain relatively efficient implementations for specific operations such as multiplication, inversion and exponentiation. Much of this work has been aimed at implementation of channel codes. The object has usually been to obtain better software implementations by using smaller tables through subfield arithmetic. Applications to hardware design have been relatively infrequent. Our techniques are directed at both hardware and software implementations. We take advantage of the efficiency obtained by the use of subfield arithmetic, not merely in the matter of smaller tables but the overall low-level (gate count). Complexity of various arithmetic operations. The computation and comparison of such gains and cost is dependent upon several parameters – the overhead of mapping between the original and the composite field representations, the nature of the underlying computation and its composition in terms of the relative frequency of various arithmetic operations, and in case of software implementations, the constraints imposed by the target architecture and its instruction set. Based on these parameters we select the appropriate field and representation to optimize a hardware circuit design. As we shall see, there can be several objectives for this optimization, such as critical

path lengths and gate counts, depending upon the overall design goals. The circuit design obtained can then be used to obtain parallelism in a software implementation by means of slicing techniques.

3.2 Proposed System:

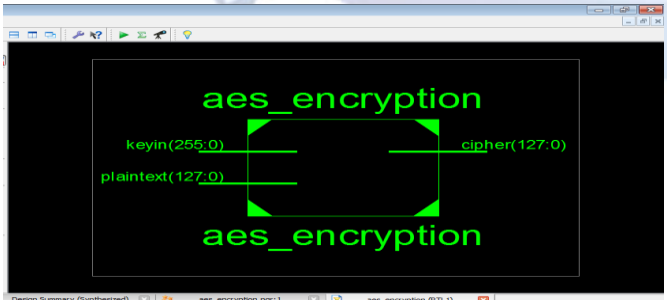
A. Parallel Advanced Encryption Standard (AES)

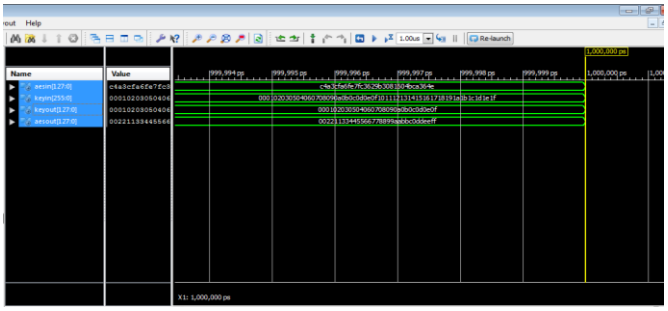
The proposed system Parallel AES is a symmetric encryption algorithm, and it takes a 128-bit data block as input and performs several rounds of transformations to generate output cipher text. Each 128-bit data block is processed in a 4-by-4 array of bytes, called the state. The round key size can be 128, 192 or 256 bits. The number of rounds repeated in the AES, Nr, is defined by the length of the round key, which is 10, 12 or 14 for key lengths of 128, 192 or 256 bits.

B. steps involved

- First Step: Sub Bytes, a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Second Step: Shift Rows, a transposition step where each row of the state is shifted cyclically a certain number of steps.
- Third Step: Mix Columns, a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- Fourth Step: Add Round Key, each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

IV. Experimental Work





V. Conclusion

We discuss three different versions of TIs of AES. We show that it is possible to achieve first-order DPA resistance with non uniform shared functions if remasking is applied properly. In the case of AES, our “non uniform” nimble implementation requires less randomness than our “uniform” raw implementation, due to the decreased number of shares. However, for other algorithms and other S-boxes, remasking may increase the amount of randomness required. This idea can be used to trade-off between the randomness and area requirements. Moreover, we empirically confirm that increasing the number of shares has a significant impact on the performance of higher-order attacks, which provides another trade-off between area and DPA resistance. Our most efficient implementation is approximately 8 k GE small and requires only 32 bits of fresh randomness per S-box calculation, which is a significant improvement over all previous works.

References

[1] Verbauwhede, P. Schaumont, and H. Kuo, “Design and Performance Testing of a 2.29 gb/s Rijndael Processor,” IEEE J.Solid-State Circuits, vol. 38, no. 3, pp. 569-572, Mar. 2003.

[2] S. Morioka and A. Satoh, “A 10-gbps full-AES Crypto Design with a Twisted BDD s-Box Architecture,” IEEE Trans. VeryLarge Scale Integration Systems, vol. 12, no. 7, pp. 686-691, July 2004.

[3] A. Hodjat and I. Verbauwhede, “Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processors,” IEEETrans. Computers, vol. 55, no. 4, pp. 366-372, Apr. 2006.

[4] S.K. Mathew, F. Sheikh, M. Kounavis, S. Gueron, A. and R.K. Krishnamurthy, “53 gbps Composite-Field AESEncrypt/DecryptAccelerator for Content-Protection in 45 nm High-Performance Microprocessors,” IEEE J. Solid-StateCircuits, vol. 46, no. 4, pp. 767-776, Apr. 2011.

[5] D.N. Truong, W.H. Cheng, T. Mohsenin, Z. Yu, A.T. Jacobson, G. Landge, M.J. Meeuwsen, A.T. Tran, Z. Xiao, E.W. Work,J.W. Webb, P. Mejia, and B.M. Baas, “A 167-Processor Computational Platform in

65 nm CMOS," IEEE J. Solid-StateCircuits, vol. 44, no. 4, pp. 1130-1144, Apr. 2009.

[6] S. Borkar, "Thousand Core Chips: A Technology Perspective," Proc. 44th Ann. Design Automation Conf., pp. 746-749,2007.

[7] A.T. Tran, D.N. Truong, and B.M. Baas, "A Reconfigurable Source-Synchronous On-Chip Network for GALS Many-CorePlatforms," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 6, pp. 897-910, June 2010.

[8] X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the AES algorithm," IEEE Trans. CircuitsSyst. II, Exp. Briefs, vol. 53, no. 10, pp. 1153–1157, Oct. 2006.

[9] S X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," IEEE Trans. Very Large Scale Integer.(VLSI) Syst., vol. 12, no. 9, pp. 957–967, Sep. 2004.

[10] A. Satoh, "A 10-gbps full-AES Crypto Design with a Twisted BDD s-Box Architecture," IEEE Trans. Very Large ScaleIntegration Systems, vol. 12, no. 7, pp. 686-691, July 2004.

[11] A. Hodjat and I. Verbauwhede, "A 21.54 gbits/s Fully Pipelined AES Processor on FPGA," Proc. IEEE 12th Ann. Symp.Field-Programmable Custom Computing Machines, pp. 308-309, Apr. 2004.

[12] J. Chang, C.-W. Huang, K.-H. Chang, Y.-C. Chen, and C.-C.Hsieh, "High Throughput 32-Bit AES Implementation in FPGA,"Proc. IEEE Asia Pacific Conf. Circuits and Systems, pp. 1806-1809, Nov. 2008.

[13] J. Granado Criado, M. Vega-Rodriguez, J. Sanchez-Perez, and J.Gomez-Pulido, "A New Methodology to Implement theAES Algorithm Using Partial and Dynamic Reconfiguration," Integration, the VLSI J., vol. 43, no. 1, pp. 72-80, 2010.

[14] S. Qu, G. Shou, Y. Hu, Z. Guo, and Z. Qian, "High Throughput, Pipelined Implementation of AES on FPGA," Proc.Int'lSymp. Information Eng. and Electronic Commerce, pp. 542-545, May 2009.