PROCEEDINGS OF SCIENCE

An Efficient Implementation of Advanced-Encryption-Standard Based on a Coarse-Grained Reconfigurable Processor

Le Chang[1]
Institute of Microelectronics of Tsinghua University,Beijing, 100084, China
E-mail: changle1901@gmail.com

Leibo Liu[2]
Institute of Microelectronics of Tsinghua University,Beijing, 100084, China
E-mail: liulb@tsinghua.edu.cn

Shouyi Yin
Institute of Microelectronics of Tsinghua University,Beijing, 100084, China

Shaojun Wei
Institute of Microelectronics of Tsinghua University,Beijing, 100084, China

Jinjiang Yang
National ASIC System Engineering Research Center, Southeast University, Nanjing, 210096,China

This paper presents a new approach to implement the Advanced Encryption Standard (AES), in the course, a Coarse-Grained Reconfigurable Processor, also called Reconfigurable Crypto-Processor is used to accomplish this assignment. The processor has been proved to be a candidate for cryptographic application. The prominent advantage of this processor is its high performance on cipher calculation tasks and flexibility in the switching of different algorithms. The method that we propose in this paper can achieve 10.47 Gbps in the 128-bits AES encryption at 300 MHz and just needs 5.01M Gates with high performance but of much more flexibility when compared with Field Programmable Gate Array (FPGA).

CENet2015
12-13 September 2015
Shanghai, China

[1]Speaker
[2]Corresponding Author

PoS(CENet2015)066

1. Introduction

A variety of electronic products such as mobile phones, PDAs, smart cards and laptops are used to store, access, manipulate and communicate sensitive data [1]. It seems that protection of the processed data has become more and more important. In most cases, we can use different security protocols by employing different cryptographic algorithms to achieve these ends.

Cryptography algorithms refer to the building blocks used to guarantee the data security. Because of the intensive nature of these algorithms, high-performance is particularly important; at meanwhile, we usually adopt more than one algorithm in the real cryptographic application. It's necessary to change the algorithm at real time [2].

Coarse-Grained Reconfigurable Arrays (CGRAs) can provide a solution to keep high-performance and flexibility at the same time. We design such a processor to meet these demands and Fig. 1 gives its architecture. It mainly consists of Compute Engine (CE) and Configuration Control (CC). When the processor works, there are primarily three flows: the configuration flow, the control flow and the data flow. The configuration information is sent to Configuration Memory (CM) as soon as the system is powered on. Reconfigurable Cell Array (RCA) is reconfigured and the data who comes from FIFO is calculated afterwards. When an algorithm ends, CC will transform RCA to different structure so that a new one can be implemented. We have mapped a lot of cryptographic algorithms thereon, such as AES [3], SM3 [4], SM4 [5] and ZUC [6].

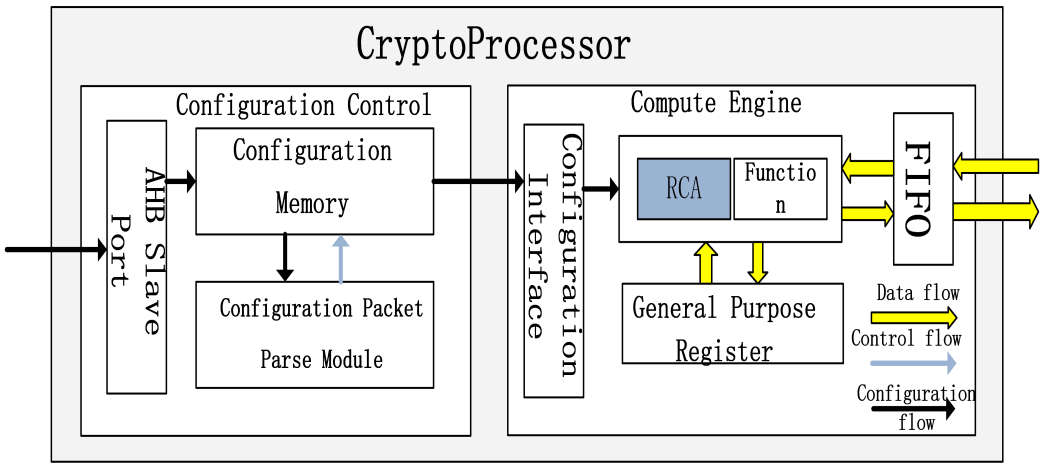

Figure 1: Architecture of Crypto-Processor

Of all these algorithms, AES is a typical block cipher and it's widely used in the network and financial system. For that reason, we map the AES algorithm on the Crypto-Processor and describe the optimization of the algorithm implementation in step by step manner. The results we obtained are compared with the implementations on the FPGAs. We will show that our mapping results can reach better efficiency.

2. Algorithm Description

The Advanced Encryption Standard, is a replacement for the Data Encryption Standard (DES). Because of insecurity of the S-Box, NIST decided to give up DES and standardize AES in May 2002.

AES is an iterated cryptographic block cypher which has a variable block length of 128, 192 or 256-bits. The number of rounds to be executed in encryption or decryption operation, depends on the block length. When the key length is 128, 196 or 256-bits, the number is 10, 12 or 14.

In this work, we focus the implementation of 128-bits AES algorithm on CGRP. It performs block cipher encryption and decryption in Electronic Codebook (ECB) mode. In the encryption process, plaintext adds the initial key, and then goes through 10 rounds function. There is no Mix Columns in Round 10, which is different from Round1-9. The decryption processes in an opposite way.

3. Algorithm Mapping

AES algorithm can be divided into two main stages: Round-Key Generation and Round Operation, both of which are mapped to the Reconfigurable Crypto-Processor in a certain way.

The Crypto-Processor we designed mainly consists of 16 lines with each line as a Programmable Entity (PE). In Fig. 2, we can see the architecture of the PE.

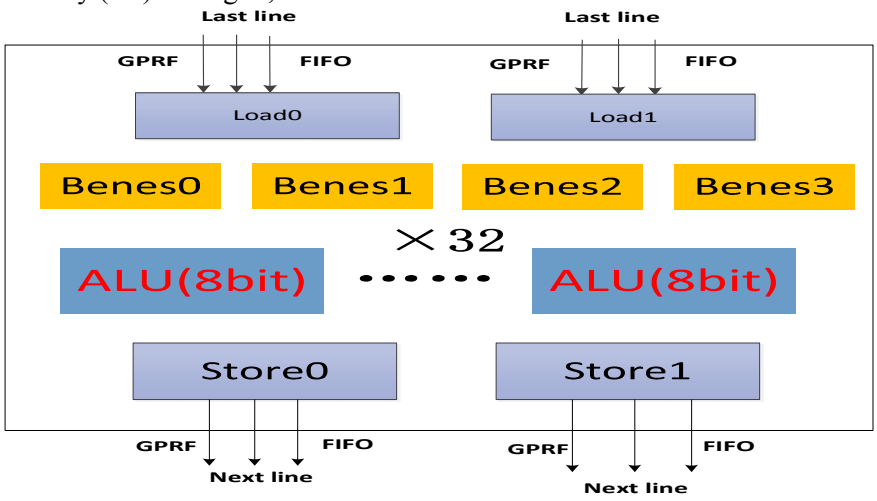

Figure 2: PE Architecture

We can swap any bit of the 256-bits data with the BENES networks. There are 32 8-bits ALUs in each PE. The four adjacent rows share one Look-up Table; therefore, we can process the SubBytes, ShiftRows, lookup tables (LUTs) or logical operations in each line. Some special operation is designed and used in the ALU. In order to reduce the use of LUTs, our models support the calculation in a finite field such as X2 operation which means $\{02\}\cdot A$ in $GF(2^8)$.

Every PE has three input ports and three output ports. The data source can be General Purpose Register File (GPRF), FIFO or the last line. The data upon computation is then sent to GPRF, FIFO or the next line. In this way, the data can be stored, exported to the processor or transferred to the PE in next line. This design can enhance the flexibility of algorithm mapping; in this sense, we can get a great ascension on performance.

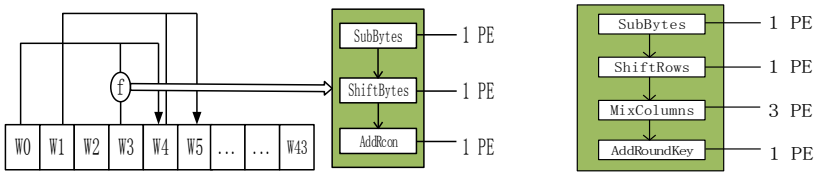

Figure 3: (a) Round-Key Generation (b) Round Operation

3.1 Key Expansion

The length of key in this paper is 128-bits. We assume that the original Round-Key is $W_3W_2W_1W_0$. Fig. 3 gives the flow path of Round-Key Generation. In this mapping method, we get a new 128-bits Key with 6 PEs.

The generation of key $W_7W_6W_5W_4$ is shown as follows:

$$W_4 = f(W_3) \oplus W_0 \tag{3.1}$$

$$W_5 = W_4 \oplus W_1 \tag{3.2}$$
$$W_6 = W_5 \oplus W_2 \tag{3.3}$$
$$W_7 = W_6 \oplus W_3 \tag{3.4}$$

According to the algorithm described in Fig. 3(a), we need to map the function F with 3 PEs. Afterwards, the generation of W_5 takes another PE. In General, 6PEs are the minimum size to calculate a new key from the old one. The new Round-Key is stored in the GPRF.

3.2 Round Operation

The round operation is the mainly time consuming part of AES algorithm and it will be remapped into the RCA upon completion of the key expansion. Fig. 3(b) shows the mapping of round operation.

It also needs 6 PEs to accomplish the mapping per round. It's obvious to implement the Sub-Bytes, ShiftRows or AddRoundKey with 1 PE. In the MixColumns, there is a transform equation:

$$S^1_{0j} = (02 \cdot S_{0j}) \oplus (03 \cdot S_{1j}) \oplus S_{2j} \oplus S_{3j} \tag{3.5}$$
$$S^1_{0j} = (02 \cdot (S_{0j} \oplus S_{1j})) \oplus S_{1j} \oplus S_{2j} \oplus S_{3j} \tag{3.6}$$

$\{02\} \cdot A$ is an operation in GF(2^8). It needs 1 PE to finish the transformation. To calculate the result of $S_{0j} \oplus S_{1j}$, 1 PE is used. The 3-input XOR operation in the end takes advantage of another PE. In the final round, 2 PEs are enough to implement it because there is no MixColumns.

The Round Operation is the main computation process in AES, and our RCA has 16 lines, that is, we can only proceed two-round operation at a time. Its performance is poor and the use ratio of the RCA is low.

4. Mapping Optimization

In our mapping scheme mentioned in Section 3, the efficiency of AES may be not optimal. It's noticed that the bandwidth of each PE is 256-bits, and we make use of half at most occasions. There is a possibility that we can improve the efficiency by analyzing the critical path and increasing parallelism.

4.1 Optimization of Key Expansion

In the key expansion, we can transform Eq.1-Eq.4 like this:

$$W_4 = Sbox(W_3 \ll 8) \oplus R_1 \oplus W_0 = Sbox(A) \oplus B \tag{4.1}$$
$$W_5 = W_4 \oplus W_1 = Sbox(A) \oplus B \oplus W_1 \tag{4.2}$$
$$W_6 = W_5 \oplus W_2 = Sbox(A) \oplus B \oplus C \tag{4.3}$$
$$W_7 = W_6 \oplus W_3 = Sbox(A) \oplus B \oplus D \tag{4.4}$$

It's obvious that the SubBytes in S-box is the critical path. Therefore we calculate B, C and D in advance. In this way, we get A, B, C and D after first PE. The values of $B \oplus W_1$, $B \oplus C$ and $B \oplus D$ are calculated in the second PE and the look-up table results of A come out in the same time. After the third PE, we get a new 128-bits key. In general, we halve the cycles from 6 to 3.

4.2 Optimization of Round Operation

In the round operation, the result after AddRoundKey is like this:

$$C^1_{0j} = S^1_{0j} \oplus K_{0j} = X2(S_{0j} \oplus S_{1j}) \oplus (S_{1j} \oplus S_{2j} \oplus S_{3j}) \oplus K_{0j} \tag{4.5}$$
$$C^1_{0j} = X2(E) \oplus (F \oplus K) = X2(E) \oplus G \tag{4.6}$$

Equation (4.6) shows that the critical path of round operation is X2 operation in GF(2^8); so we divide the MixColumns into three parts to recombine them with less PEs. In the first part we calculate E and F; at meanwhile, it can process ShiftRows. In the second part, we get the X2(E)

and G. It means that we add the round key in advance. The last part we'll see the final result of current round. Upon optimization, we reduce the number of PE from 6 to 4.

5. Simulation Results and Comparison

In this section we compare the performance of AES running on our Crypto-Reconfigurable-Processor and on several FPGAs [7, 8]. Our design is verified under SMIC 55 nm CMOS. We implement the optimized AES mapping thereon practically and obtain the performance. Table 1 gives the comparison result.

Source	Frequency (MHz)	Technology (nm)	Throughput (Gbps)	Resource	Area efficiency (bps/Gate)
W. Yi, 2013	319	40	40.9	9097 slices	89
H. Li, 2012	92	65	0.977	1015 slices	49.6
This paper	300	55	3.05	5.01MGates	608.5
Thispaper (optimized)	300	55	10.47	5.01MGates	2089

Table 1 : Comparison with Some FPGA Implementations

According to the Table 1, we can see that it achieves higher area efficiency than FPGA implementations. Upon optimization, the throughput of AES can increase twice. At the same time, we implement the SM3, SM4 and ZUC algorithms in an acceptable performance. The SM3 implementation can achieve 216Mbps, SM4 reaches 2.34Gbps and ZUC comes to 1.2Gbps; thus we can switch the algorithms in real-time easily with high throughput under the practical working condition.

6. Conclusion

To sum up, this paper proposed a new implementation of AES based on reconfigurable Crypto-Processor. The performance of this implementation is higher than FPGA. At the same time, the processor's more flexible because it can support the real-time reconfiguration. Up to now, we have implemented AES, SM3, SM4, ZUC and SHA-256. We will implement other algorithms and improve the architecture in future.

References

[1] Garcia, A., Berekovic, M., & Vander Aa, T. (2008, July). *Mapping of the AES cryptographic algorithm on a coarse-grain reconfigurable array processor.* In Application-Specific Systems, Architectures and Processors, 2008. ASAP 2008. International Conference on (pp. 245-250). IEEE.

[2] Bossuet, L., Grand, M., Gaspar, L., Fischer, V., & Gogniat, G. (2013). *Architectures of flexible symmetric key crypto engines—a survey: From hardware coprocessor to multi-crypto-processor system on chip.* ACM Computing Surveys (CSUR), 45(4), 41.

[3] Pub, N. F. (2001). *197: Advanced encryption standard (AES).* Federal Information Processing Standards Publication, 197, 441-0311.

[4] Tianyong, A., Zhangqing, H., Kui, D., & Xuecheng, Z. (2014, April). *A compact hardware implementation of SM3*. In Consumer Electronics-China, 2014 IEEE International Conference on (pp. 1-4). IEEE.

[5] Cheng, H., & Ding, Q. (2012, December). *Overview of the block cipher.* In Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on (pp. 1628-1631). IEEE.

[6] Kitsos, P., Sklavos, N., & Skodras, A. N. (2011, August). *An FPGA implementation of the ZUC stream cipher.* In Digital System Design (DSD), 2011 14th Euromicro Conference on (pp. 814-817). IEEE.

[7] Wang, Y., & Ha, Y. (2013). *FPGA-based 40.9-Gbits/s masked AES with area optimization for storage area network*. Circuits and Systems II: Express Briefs, IEEE Transactions on, 60(1), 36-40.

[8] Li, H., Ding, J., & Pan, Y. (2012). *Cell array reconfigurable architecture for high-efficiency AES system*. Microelectronics Reliability, 52(11), 2829-2836.