Hardware and Software Co-Design: Application of the Technique in the Implementation of the AES Cryptography Algorithm

Lucidia Assunção Silveira
UNIPAMPA – Federal University of Pampa
Bagé-RS, Brazil
lucidiasilvera@gmail.com

Maurício Realan Arrieira
UNIPAMPA – Federal University of Pampa
Bagé-RS, Brazil
mauriciorealan@gmail.com

Bruno Silveira Neves
UNIPAMPA – Federal University of Pampa
Bagé-RS, Brazil
brunoneves@unipampa.edu.br

Fábio Livi Ramos
UNIPAMPA – Federal University of Pampa
Bagé-RS, Brazil
fabioramos@unipampa.edu.br

ABSTRACT

This article intends to demonstrate the efficiency of using the hardware and software co-design technique, through the implementation of the AES encryption algorithm. Hence, the MixColumns step (which is a part of the AES algorithm) was implemented in hardware, to demonstrate that the integration of hardware and software leads to a performance gain, when compared to a solution purely implemented in software. Thus, the present work demonstrates the potential gains by using the co-design strategies for embedded systems and FPGAs design.

KEYWORDS

AES, Co-Design, Cryptography, MixColumns.

1 INTRODUCTION

The interest in cryptography methods for embedded systems has been growing recently, especially when it comes to mobile devices [1]. This phenomenon happens because this type of device usually presents many limitations. Although it is common to use algorithms that are already famous for the use on computers or high-capacity servers, the devices often do not have the capacity to execute them, since these algorithms require a lot of processing [2].

Throughout the history of information security, several cryptographic algorithms have been used to protect data in different categories. Currently, the most widely used algorithm in the industry, being also the most cited in the academic environment, is the AES (Advanced Encryption Standard). The AES algorithm is well known not only for providing a high level of data security, but also for its high computational speed during the process of encryption/decryption of the information and its low memory consumption, which enables its wide use on mobile devices [3].

It is increasingly necessary to have secure exchange of electronic information,, which the AES is a suitable option for that purpose. The AES algorithm also needs to be used in low cost products such as wireless devices, cell phones and many other embedded systems. Thus, the implementation of the algorithm should be less and less costly [3].

Embedded systems typically are the union of hardware and software, the hardware being units for specific applications and software a program running on a general purpose hardware unit. Typically, software is cheaper, but slower, while hardware is more expensive but more efficient. Therefore, choosing which parts to implement in hardware and which ones will be built in software is very important to obtain a good performance [4].

For embedded systems, in order to achieve optimization, the use of co-design is strongly indicated, since this method exploits the interactions between hardware and software, because when both are developed together, the system will only contain the characteristics necessary for its correct operation [5].

In this work, the AES algorithm is implemented in different contexts to explore the communication between software and hardware. The co-design approach was used, in order to evaluate the performance of the algorithm in this situation.

In this way, after this introduction section, in section 2 will be presented a theoretical reference, where relevant concepts are presented in the understanding of the theme. Then, in section 3, the methodology adopted for the project will be presented. In section 4, the implementation of the proposed solutions is described, and later in section 5 the results are presented. Finally, brief conclusions will be explained and developed.

2 THEORETICAL REFERENTIAL

2.1 AES - Advanced Encryption Standard

The AES algorithm, also called Rijndael, is one of the most widely used encryption algorithms in the world. A wide range of computing devices make use of it [6]. In addition to its security properties, it is also a very efficient algorithm on several platforms, like 8-bit microprocessors, 64-bit processors and also FPGAs

(Field Programmable Gate Array). Its efficiency was a crucial factor for this algorithm to become a cryptographic standard [7].

AES is a fixed block implementation of the Rijndael algorithm, with supports 128, 192 and 256-key bits. The current state array, that is, the content being encrypted, is organized into a 4x4 array, which is transformed N times by a function. The number of spins can be 10, 12 or 14, depending on the size of the encryption key set.

To perform the encryption process, the state is initialized and then a 128-bits xor and state is performed. The state is modified N-1 times until the last run, which is slightly different. In each step, four functions are realized: SubBytes, ShiftRows, MixColumns e AddRoundKey. In the last step, however, the MixColumns function does not run. Each function operates in the state, at each round, as follows [8]: *(i)* SubBytes: Replaces each byte with an entry in the S-Box table, which is a pre-calculated substitution table; *(ii)* ShiftRows: The lines i of the matrix are shifted to the left, cyclically, being $0 \leq i \leq 3$; *(iii)* MixColumns: Multiplies each column, taking as a polynomial of order less than 4 with coefficients greater than 2^8 in module $x^4 + 1$ by a fixed polynomial; *(iv)* AddRoundKey: make a xor of the key of the round *r* with the state.

2.2 Co-Design - Hardware/Software

The term hardware and software co-design emerged in the early 1990s to describe a confluence of problems in integrated circuits [9]. The co-design attempts to increase the predictability of the embedded system design by providing analysis methods that tell the designer if the system meets performance, power and size requirements; and synthesis methods that allow designers to quickly evolve the project methods.

Most electronic systems, whether embedded or not, have a digital component that is nothing more than a hardware platform that runs software applications. The co-design would be the exploitation of the hardware and software synergism through its parallelism, to obtain the system objectives and constraints [10].

In hardware and software co-design, designers consider manipulations in a fashion that software and hardware components of a system work together to achieve a certain desired behavior, in accordance with performance objectives and certain implementation technologies. One type of co-design aims to accelerate software applications by extracting portions of code and those implemented in hardware. This acceleration of software can occur even in general-purpose computing, by using programmable hardware (e.g. FPGA) [11].

2.3 Related Work

The work of [12] highlights the Data Encryption Standard (DES), encryption algorithm and its description in VHDL. The Data Encryption Standard (DES) - is a cryptographic standard created in 1977 through a bid open by the former National Security Agency (NSA). DES is the pre-AES encryption standard. It is proposed as a cryptoprocessor project that can be divided into four main phases: *(i)* the algorithmic description in VHDL of the main cryptographic algorithms; *(ii)* implementation in software, using the C language; *(iii)* the design, the description in VHDL and the implementation in FPGAs of the cryptoprocessor and finally; *(iv)* the cryptoprocessor performance analysis.

The work of [13] shows the importance and feasibility of building cryptographic systems in hardware, discussing the aspects that contribute to it, as well as showing a way to achieve this goal. Therefore, it was concluded that the cryptographic implementation in hardware through FPGAs is desirable, being faster and, considering the size of the application, even cheaper to perform. The VHDL is also a language that allows facilities for the design of projects for reconfigurable circuits, being very useful for the accomplishment of cryptography through them.

An alternative implementation an ASIC AES encryption algorithm is presented in [14]. Thus, it indicates that the use of hardware provides adequate performance for several implementations, in which the use of software does not meet the necessary requirements. An architecture model was developed looking for a balance between area surface reduction, high performance and low power consumption.

3 METHODOLOGY

The methodology used at this worj consists of the following activities: *(i)* definition of the hardware and software co-design strategy as the focus of study; *(ii)* later, the AES algorithm was defined as an algorithm to be developed from the application of co-project strategies; *(iii)* selection of tools for implementation, validation and evaluation of the solution, based on co-design strategies; *(iv)* definition of the MixColumns module as the step to be implemented in hardware; *(v)* in the fifth step, the execution of the algorithm partitioning, into software and hardware components; *(vi)* implementation of co-synthesis followed by integration of components; *(vii)* the validation of the new solution proposal and also the verification and evaluation of its relation of area and operation; *(viii)* Finally, in the last step, an analysis of the results obtained is carried out.

For the implementation of the project were defined the following tools:

- Altera's Qsys environment as an interconnection tool for the implementation of hardware and software components of the AES algorithm;
- Nios II Embedded kit for prototyping, which is based on a programmable device (FPGA);
- Cyclone III FPGA, from Altera, was chosen to be used;
- Altera Quartus tool was defined as a tool for describing the hardware in VHDL;
- The Eclipse tool to execute AES implemented in software, developed in the C programming language.

In this context, Fig. 1 presents, in an illustrative way, the communication between the tools adopted in the implementation of the algorithm with co-design (i.e. the application for software development, the interconnection environment, the hardware description tool and the FPGA).

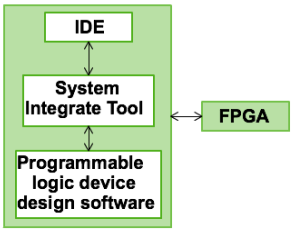

Figure 1: Communication of Tools Used.

Finally, it is necessary to emphasize that the choice of the MixColumns stage as a hardware modulus was made due to the complexity of this step, being the AES module that presents a structure of intermediate complexity when compared to the other stages of the AES. Thus, it was verified that this step is equivalent to 8% of the execution of the algorithm.

4 IMPLEMENTATION AND DEVELOPMENT

In this section, a sequential implementation will be presented, as well as a semi-combinational solution and, finally, a description of the communication between hardware and software. As one may notice, the implementations of the MixColumns module in hardware were developed through the VHDL hardware description.

4.1 Sequential Solution

The implementation of the AES algorithm in MixColumns module hardware was developed primarily in a sequential fashion. This solution makes use of registers, which store the input values, the results of partial operations and the output values.

The computation of the whole operation is performed in stages, where each stage performs a different activity. The actions taken at each stage are: (i) at the first stage, the entries are stored in registers; (ii) at stage two, bit shifts are made to the left in each of the registers created at the previous stage; (iii) at the third stage, some constants are analyzed and defined to be used in the next State; (iv) at stage four, XOR operations are performed, among the elements registered at the first stage and at the second stage, and stored with the constants defined at stage three; (v) at the fifth and last stage, the values stored at stage four are sent to the output. The structure of this solution is shown in Fig. 2.

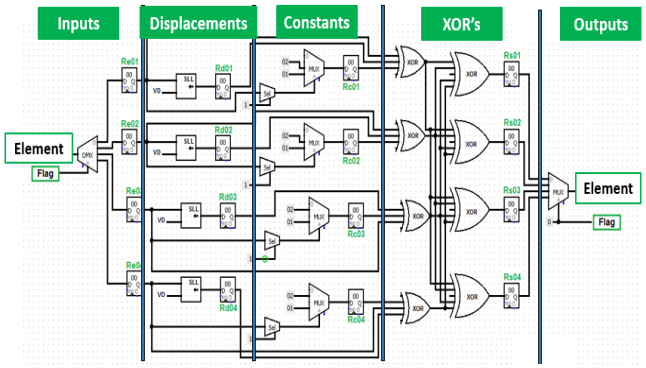

Figure 2: Sequential Solution Implementation Framework.

4.2 Semi-Combinational Solution

A semi-combinational MixColumns module solution of the AES algorithm was also developed. This solution, unlike sequential solution, uses only registers to store the input and output values (i.e. the internal stages are processed in a purely combinational fashion). Therefore, the computation of the whole operation is accomplished with fewer stages than in the sequential solution. Hence, after the registering of the entries, all operations (i.e. displacements, constants derivation and XOR operations) are performed within one cycle, until the results are sent to the output. In this way, Fig. 3 illustrates the stages that are part of the proposed solution.

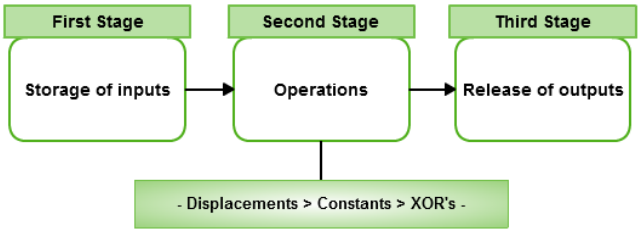

Figure 3: Semi-Combinational Solution Projected.

4.3 Communication Hardware - Software

The logic of communication between software and hardware was implemented right after the MixColumms hardware design. In this process, the software sends to the hardware two elements, a variable that makes the control of the element that is being sent, and an element of the array of partial states of the AES algorithm. The hardware, in turn, sends an updated software control variable and an element that has been computed by the MixColumns operation.

The software sends a hardware element of the array of partial states at a time, using the control variable to maintain the correctness of the element that is being sent. After sending four elements, all from the same column of the matrix, the software sends the control variable, which indicates the end of the column. Hence, the hardware performs the MixColumns operation and subsequently sends an output element at a time to the software. This operation is repeated four times, exactly for the computation of the four columns of the matrix.

5 RESULTS AND DISCUSSIONS

Tests were carried out on the two solutions developed, presented in the previous section. The computer used in the development of the project and the execution of the tests is a machine with Intel Pentium Quad Core N3700 processor and Windows 10 operating system.

At first was conducted a test on purely software AES algorithm, that is, the AES algorithm in the C programming language, running on the NiosII processor. In order to perform these tests, an integration between the Eclipse tool (for C software) and Quartus software (for the hardware) was used, whereas the communication was performed by the Qsys environment.

Later tests were carried out on the proposed solutions that utilize the co-project strategy. A comparison was drawn between the runtimes of the pure software solution (the baseline for comparision), semi-combinational solution, and sequential solution solution, all running on the Cyclone III FPGA and NiosII processor.

Therefore, these results can be observed in Table 1. The execution times were obtained through the communication terminal of the Eclipse tool, which indicates the total time spent in the complete execution of the algorithm, both of co-design solutions running all AES algorithm steps in software, except by the MixColumns module, which was running in the FPGA.

Table 1: Runtimes of Implementations.

Implementation	Runtime (*us*)
Pure Software	317108
MixColumns in Sequential Hardware	305487
MixColumns in Semi-Combinational Hardware	305307

It can be noticed, examining the results obtained in the tests, that the semi-combinational solution proved to be the most efficient among the implementations. It is possible to analyze that the difference between the implementation of the sequential and the semi-combinational solution is not very large, which is in conformity with the expected difference for the implementations, due to a solution run in three cycles and another in five.

The semi-combinational solution showed a gain of approximately 3.72% in relation to pure software implemantation. Sequential solution presented a gain of approximately 3.68% compared to pure software implementation. These results are according the expected, since the runtime portion of the MixColumns module in the total software implementation of the AES algorithm is approximately 8%. This value of 8% was obtained through tests performed in the algorithm to find out how much each operation module occupies in the total execution of the algorithm.

Another analysis is about the area that each co-design solution occupies in the FPGA device. It is necessary to point out that the hardware description covers more than just the implementation of the solution, but also the entire interconnection project between the software and hardware environment Qsys. The area results can be seen in Table 2.

Table 2: The Area Consumed by the FPG Implementations.

Implementation	Total Area Consumed	Area Consumed by MixColumns Module
Sequential	11% of the FPGA	3% of the FPGA 146 Logical Elements
Semi-Combinational	10% of the FPGA	2% of the FPGA 131 Logical Elements

Therefore, we can observe that the semi-combinational solution consumes less FPGA area than the sequential solution. This occurs both in the analysis of the project as a whole, and in the analysis of only the MixColumns modules. This is mainly due to the reduction in the number of intermediate registers.

6 CONCLUSION

The results obtained, particularly in respect of the runtime, have proved the improvement potential of co-design project for the AES algorithm. Moreover, the AEC is a valid evaluation of the co-design implementations, since this approach has demonstrated better performance than the algorithm implementation in purely in software. Furthermore, the co-design approach can be used for other types of algorithms, and that when implemented correctly, it tends to present a significant performance gain, depending on the operation implemented in hardware.

For optimum use of the strategy of co-design fashion, for the AES algorithm, maybe the implementation of one of the other three stages might be more efficient, since these occur in all rounds, unlike the MixColumns, that does not run on last stage. In addition, it was possible to identify that, as expected, sequential logic uses more resources than semi-combinational logic.

REFERENCES

[1] Kocher, P., Lee, R., McGraw, G., Raghunathan, A., and Ravi, S. (2004). Security as a new dimension in embedded system design. Symposium on Design Automation and Microprocessors (DAC).

[2] Silva, Thiago H., Douglas G. Macharet, and César F. Teixeira. "Análise do desempenho de algoritmos criptográficos em dispositivos móveis." *Wperformance Workshop de Desempenho de Sistemas Computacionais e de Comunicação, Anais do XXVIII Congresso da SBC, Belém*. 2008.

[3] Chodowiec, Paweł, and Kris Gaj. "Very compact FPGA implementation of the AES algorithm." *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2003.

[4] Mann, Zoltán Ádám, András Orbán, and Péter Arató. "Finding optimal hardware/software partitions." Formal Methods in System Design 31.3 (2007): 241-263.

[5] Gupta, R. and Micheli, G. D. (1993). Hardware-software co-synthesis for digital systems. In IEEE Design & Test of Computers, pages 29–41.

[6] Lu, Chih-Chung, and Shau-Yin Tseng. "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter." Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on. IEEE, 2002.

[7] Osvik, Dag Arne, et al. "Fast software AES encryption." *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2010.

[8] Tonde, Ashwini R., and Akshay P. Dhande. "Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA." *International Journal of Current Engineering and Technology* 4.2 (2014).

[9] Wolf, Wayne. "A decade of hardware/software codesign." *Computer* 36.4 (2003): 38-43.

[10] De Michell, G., and Rajesh K. Gupta. "Hardware/software co-design." Proceedings of the IEEE 85.3 (1997): 349-365.

[11] Thomas, Donald E., Jay K. Adams, and Herman Schmit. "A model and methodology for hardware-software codesign." IEEE Design & test of computers 10.3 (1993): 6-15.

[12] Pereira, Fábio Dacêncio, and Edward David Moreno. "Otimização em VHDL e Desempenho em FPGAs do Algoritmo de Criptografia DES." *Quarto Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD), São Paulo*. 2003.

[13] Almeida, Ariane A., and Vaston G. da Costa. "Criptografia em Hardware com VHDL Usando Circuitos FPGA× Criptografia em Software." (2010)

[14] Girardi, Alessandro, et al., “Um Hardware ip para criptografia no padrão AES”.