Design and SystemC Implementation of a Crypto Processor for AES and DES Algorithms

M. Aşkar, T.Egemen, *Middle East Technical University*

***Abstract*— In this study, the design and SystemC implementation of a programmable Crypto Processor for Advanced Encryption Standard (AES) and Data Encryption Standard (DES) algorithms is presented. Both of the algorithms are implemented in a single architecture instead of using separate architectures for each of the algorithm.**

The design of the proposed architecture is studied using SystemC tool. The architecture is implemented in modules with using the advantages of SystemC [17]. The simulation results from SystemC are used to verify the proposed design. Then SystemCrafter tool [18] is used to compile the SystemC descriptions into hardware.

***Index Terms*— Advanced Encryption Standard (AES), Bit Permutation, Data Encryption Standard (DES), Encryption**

I. INTRODUCTION

GENERALLY the designs of encryption and decryption are implemented for only a specific algorithm. It is easy to implement the hardware for a single algorithm. But with such designs, it is not possible to process different crypto algorithms.

In this work a programmable Crypto processor is implemented for AES [3], DES [4] and TDES [5] by using SystemC tool. The architecture is similar to that of the general microcontroller's structure, but there are some additional specifications. A Permutation module is added to the architecture, which is responsible for all of the bit permutation operations.

A new Instruction Set Architecture (ISA) is proposed for this design and Encryption - Decryption algorithms of AES, DES and TDES are implemented by using these instructions. The internal structure of the implemented architecture is based on 32-bit data for small area realization. DES, TDES and all the key and data lengths of AES algorithms are supported with this implemented architecture.

II. CHARACTERISTICS OF AES AND DES

AES algorithm is an iterative block cipher performing encryption and decryption in fixed size blocks. The incoming data and key are stored in a matrix, called state matrix, and all operations are performed over the state matrix [1] [2]. There are three different input lengths for data and key length, which are 128, 192, 256 bits. Each iteration is called a round and the round number is changed depending on the data and key length. Byte Substitution, Shift Row, Mix Column and Add Round Key transformations are the four main transformations in one round [3]. In Byte Substitution the State byte is replaced with a substitution table element, which is calculated with nonlinear transformations in GF (2^8). In Shift Row, rows of the State matrix are shifted to the right cyclically. For each data length and for each State matrix row, there is a different shift offset value. Mix Column transformation acts independently on every column of the state. Each column of the State matrix is considered as a four-term polynomial over GF (2^8) and multiplied with a fixed polynomial. Add Round Key is applied to the State by a simple bitwise EXOR operation. Decryption process is the inverse operation of the encryption process and the transformations in the encryption round are also reversed in the mean of the sequence.

DES is a symmetric crypto algorithm, which operates on 64-bit block size within 16 rounds. The input plain text and the output ciphered text are 64-bit. The input key data length is also 64-bit, but only the 56 bits of the whole key data is effective [4]. The remaining 8 bits have no effect on the encryption/decryption process of the DES. The main operations are bit permutations and substitution in one round of DES. There are six different permutation operations, which are used both in Key Expansion part and cipher part. The main operations in DES algorithm, like key related operations and SBox operations are performed in the Cipher part. Decryption of DES algorithm is similar like encryption, but only the round keys are applied in reverse order.

III. SYSTEM ARCHITECTURE

For AES, DES and TDES algorithms implementation, the basic transformations are implemented in a processor structure. The processor architecture is given in Figure 1 and it is based on a combination of different modules, which are given below.

Murat Aşkar, Tufan Egemen, *Middle East Technical University*
Electrical and Electronics Engineering Dept. Middle East Technical University, askar@eee.metu.edu.tr, *tufan.egemen@beko.com.tr*

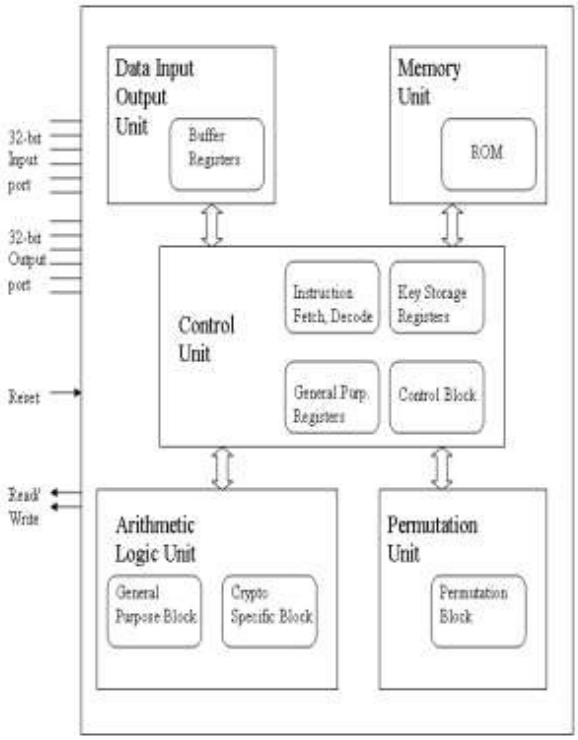

Figure 1. The main crypto processor architecture diagram

--Arithmetic Logic Unit (ALU) Module
--Permutation Module
--Control Unit Module
--Data Input Output Module
--Memory Module

A. Control Unit Module

Control Unit is the main module of the architecture. The main function of this module is to control the others modules activities according to the instructions stored in memory. First of all the instructions are fetched from memory and decoded for further operations. After the decode process of the instruction the execution process is activated. In this process the Control Unit sends the proper control signal to the related modules with the proper operand data. Then the result data is copied back to related registers to finish one instruction operation.

One machine cycle in the implemented architecture is consisting of 4 clock cycles. During a machine cycle fetch, decode and execute operations are performed. The most of the instructions are one machine cycle instructions, but some of the instructions are processed in three or four cycles, like shift operations in AES algorithm.

The State values of the algorithms are stored in 32-bit registers after the processes [13]. There are used eight 32-bit register to store all the State values. There is also a different register block in the Control Unit module, which stores all the round key values. There are 120 32-bit registers in this register block to supply all different round key values for both of AES, DES ad TDES.

B. Data Input/Output Module

The external access to the implemented Crypto processor is provided by the Data Input/Output Module. The Data I/O module has two different 32 bits external interface. One of the interfaces is assigned as input to the processor and the other one is assigned as output. The input and output data are stored initially in buffer registers, than moved to the internal registers or moved to output.

C. Memory Module

The Memory Module is consisting of a ROM block. The instructions are stored in the ROM block and they are subjected to the Control Unit with an 8-bit wide data link between Control Unit and Memory Module.

D. Arithmetic Logic Unit (ALU) Module

ALU Module is responsible of processing the incoming data according to the commands of Control Unit. The basic operations like Boolean functions, addition – subtraction operations, shift operations are performed in the implemented ALU module. Further to that some AES and DES specific operations like SBox, Mix Column are also performed in the ALU module.

There are only one ALU module in the implemented architecture in respect of two ALU unit [7] or four functional unit applications [8]. The main operations are performed over 32 bits data, but there are some exceptions for both of AES and DES algorithms. Some operations are performed over bytes and in those cases the incoming 32 bits data is divided into suitable data chunks and then the operations are performed.

The SBox operations for both of AES and DES algorithms are performed by using Look up Tables. There is one Look up Table structure, which contains all the SBox values for both of AES and DES. A memory element in the Look up Table is 8-bit wide and it is convenient for AES algorithm.

But in DES algorithm SBox operation there are eight different SBox table and their outputs are 4-bit data. For unified Look up Table structure, the sequential two SBox values in each DES SBox table are stored in the same memory address, which is given in Figure 2.

The SBox Look up Table can be reprogrammed for any other purpose. The 256 memory element of the Look up Table can be reloaded for different applications [9]. For example the AES and DES algorithm can be performed with only one Look up table. The memory elements can be reconfigured depending on the crypto algorithm.

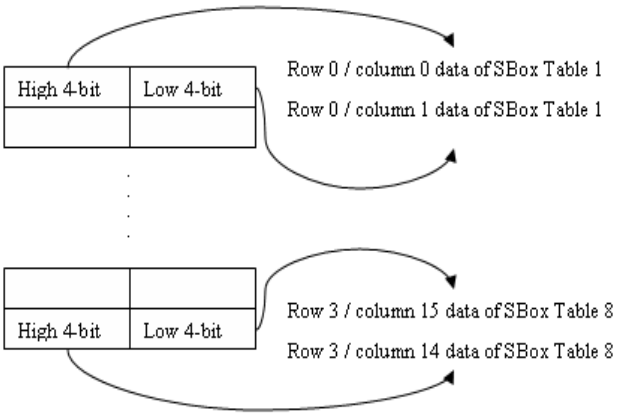

Figure 2. DES Look Up Table structure

E. Permutation Module

In the bit permutation operations, the incoming data is subjected to the some bit position changes according to the permutation type, to improve diffusing properties. The bit permutation operations have a big process part in DES and TDES algorithms. In many other solutions for DES algorithm these blocks are mainly implemented as look up tables or implemented as hardware routing. In this architecture a separate permutation module is implemented for bit permutations of DES and TDES. The main structure of Permutation Module is based on Benes network structure, which is a combination of a Butterfly network and Inverse Butterfly network [15][16]. The Benes network structure is given for 8-bit data permutation in Figure 3.

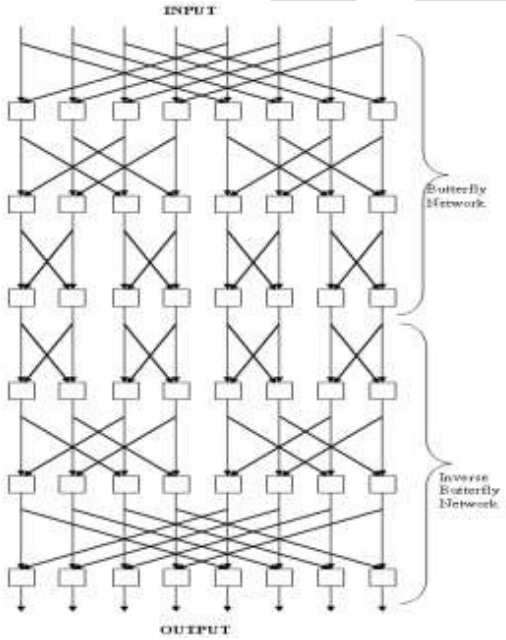

Figure 3. Benes Network structure for 8-bit permutation

The implemented module is designed for permutations of 64 bits data. Therefore there are used 12 stages in the module for permute 64-bit data. The half of the total 12 stages belong to Butterfly network and the remaining half belong to Inverse Butterfly network. For each permutation there are used a different set of control bits to control the data flow between sequential stages. There are used 12 different control bit register, each one is 32 bits. Permutation Module is also able to permute any 64-bit data. For this purpose only the control bit registers should be reloaded with the new control data according to the desired permutation by using the proper instructions in the ISA.

F. Instruction Set Architecture

Table 1 gives some examples instruction of the Instruction Set Architecture for implemented crypto processor design.

TABLE I
ISA overview of implemented design

Instruction	Operation	Syntax	Opcode	Machine Cycle
XTME	xtime operation on Mix Column operation of AES algorithm	*xtme ri* *i = 0 .. 7*	0x40 to 0x47	1
MIX	exor operations after xtime instr. in Mix Column	*mix ri , rj* *i = 0 .. 7* *j = i + 8*	0x48 to 0x4f	1
SBOX	sbox operation for both of AES and DES algorithms	*sbox ri* *i = 0 .. 7*	0x30 to 0x37	1
SHFT	AES 128-bit Shift Row operation	*shft 128*	0x38	3
SHFT	AES 192-bit Shift Row operation	*shft 192*	0x39	5
SHFT	AES 256-bit Shift Row operation	*shft 256*	0x3a	5
EXOR	Bitwise EXOR operation	*exor ri , rj* *i = 1 .. 7* *j = i - 1*	0x24 to 0x2a	1
MOV1	Store accumulator group1 values	*mov1*	0x50	1
MOV2	Store accumulator group1 values	*mov2*	0x51	1
EXK0	Key exor operation	*exk0 #r_number* *r_number = 1 ..f*	0x60 to 0x6f	2
EXK1	Key exor operation	*exk1 #r_number* *r_number = 1 ..f*	0x70 to 0x7f	2
RCON	Load Rcon values to reg0	*rcon* *#r_number* *r_number = 1 ..f*	0xf0	1
PC1P	DES Permute Key PC1	*pc1p r0 , r1*	0xb0	1
PC2P	DES Permute Key PC2	*pc2p r0 , r1*	0xb1	1
INIP	DES Initial Permutation	*inip r0 , r1*	0xb2	1
ETBP	DES E Table Permutation	*etbp r0 , r1*	0xb3	1
RORB	Rotate byte to right	*rorb r0*	0xa0	1
RORD	Rotate bit to right in DES	*rord r0 , r1*	0xa1	1

All the instructions in the ISA are 8 bits in length. Most of the instructions have two input operands and one output operand like general microprocessors. But there are also some instructions, which have one input operand to supply the crypto algorithm specification.

The ISA consist of mostly one machine cycle instructions like SBox, Permute, but there are also some instructions, which are more than one machine cycle.

IV. IMPLEMENTATION RESULTS

The design is implemented using SystemC and the main simulations are based on the outputs of the SystemC tool. The simulation results are analyzed to verify the implemented architecture. The encryption and decryption algorithms for both of the AES, DES and TDES are simulated for different data and key lengths. Table 2 gives the machine cycle values of the implemented Crypto processor for the related algorithms.

TABLE II
Machine Cycles for performed Crypto Algorithms

Crypto Algorithm	Machine Cycle
128 AES	153
192 AES	239
256 AES	369
DES	131
TDES	393

Some blocks of ALU Module and Permutation Module are compiled using SystemCrafter tool to hardware, and then synthesized with Xilinx tool to Spartan3AXC3S200A device. The results according this process are given in Table 3.

TABLE III
Slice values for some crypto specific blocks

Crypto Specific Block	Number of occupied Slices
Xtime (Mix Column in AES)	63
Xtime2 (Inv. Mix Column)	182
Shift128 (128-bit AES)	197
Shift192 (192-bit AES)	215
Shift256 (256-bit AES)	227
Mix (Mix Column in AES)	178
Rotate Byte	60
Rotate Bit (DES key expansion)	94
Permutation	1672

V. RELATED WORK

Generally the designs for crypto algorithms are dedicated to a specific algorithm. In such kind of designs it is easy to configure the hardware to get the desired specification [6][10]. Therefore in these designs the crypto algorithm process is quite fast.

For the main parameters of a chip the area and the throughput can be adjustable to a specific application. For example to get the maximum throughput, all the iterated rounds of the algorithm can be implemented sequentially with an inner-round and outer-round pipeline structure [12]. In other case only one iterated round can be implemented with proper internal data length parameters for an area constraint design. Besides the algorithm specific design, there are also some programmable architectures, which can perform several crypto algorithms.

A study reported in [7] is a programmable Crypto Processor architecture, called Cryptonite, which is able to perform DES and 3DES, AES, IDEA, RC6, MD5, and SHA-1 algorithms. This architecture has a different instruction set for cryptographic processing such as parallel 8-way permutation lookups, parameterized 64-bit/32-bit rotation, and a set of XOR-based fold operations.

These instructions are for the core functions of different crypto algorithms and shows differences than general purpose instructions. All instructions are executed in a single cycle. 64-bit and 32-bit computation supported in this study. The results of [7] are given in Table 4.

TABLE IV
Cryptonite Architecture Results

Algorithm	Throughput (Mbit/s)	Cycle count	Speed (MHz)
DES	732	35	400
TDES	244	105	400
AES	731	70	400

Another architecture for the programmable processor is reported in [8]. The presented architecture, called CryptoManiac, is a 4-wide, 4-stage 32-bit VLIW processor with a three input operand ISA, and it is able to perform TDES, IDEA, RC6, AES and Twofish algorithms. There are four parallel functional units in the CryptoManiac architecture. The process in the architecture is started with fetching a single VLIW instruction word that contains four independent instructions. The instruction set consist of 32-bit instructions and enhanced for the cryptographic processes by combining general arithmetic instructions with logical instructions, substitutions with logical instructions, and rotate operations with logical instructions. The results related to [8] are given in Table 5.

TABLE V
CryptoManiac Architecture Results (Estimated values in [7])

Algorithm	Throughput (Mbit/s)	Cycle count	Speed (MHz)
TDES	68	336	360
AES/128	90	511	360

VI. CONCLUSION

This study presents a programmable Crypto Processor for AES, DES and TDES algorithms containing both encryption and decryption processes in the same design for all data and key lengths. A new Instruction Set Architecture is suggested and implemented to easily process all the different modes.

The hardware architecture of this design is implemented using SystemC. The main architecture is divided into modules and each module is implemented separately. The main parts of ALU module and Permutation Module in SystemC descriptions, which are related with the basic transformations of the crypto algorithms, are compiled into hardware using the SystemCrafter tool. And then the outputs of the SystemCrafter tool are used in synthesis process together with Xilinx tools.

For the implemented Crypto Processor design, 32-bit architecture is proposed. The main reason for the 32-bit architecture is, implementing an area efficient design for small area applications. As a result of this architecture, the total process time of the performed algorithm will be longer than other structures, which are based on 64-bit or higher data lengths.

DES algorithm can be performed using 21 different instructions with the proposed ISA. On the other hand AES-128 algorithm can be performed using 32 different instructions. There are 9 common instructions like SBOX, EXOR and MVK0 (store round key values) in the ISA, which are used for both of the DES and AES algorithms. Therefore it is clear that implementing AES and DES algorithms in a single design is an efficient way to decrease the area.

Finally, the Permutation Module is also an important part of the implemented design. In general applications, the bit permutation operation is implemented in memory based structures or as hardware routing structure, which are dedicated to only one permutation. But all of the bit permutation operations of DES and any other 64-bit permutation can be performed in a single structure with the implemented Permutation Module.

REFERENCES

[1] The Design of Rijndael AES – The Advanced Encryption Standard, John Daemen and Vincent Rijmen, Springer-Verlag, 2002.
[2] AES Proposal: Rijndael, John Daemen and Vincent Rijmen, September 3, 1999.
[3] Advanced Encryption Standard (AES). Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[4] Data Encryption Standard (DES). Available: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
[5] National Institute of Standards and Technology Computer Security Resource Center: http://csrc.nist.gov/
[6] Refik Sever, A. Neslin İsmailoğlu, Yusuf C. Tekmen and Murat Aşkar, "*A High Speed ASIC Implementation of The Rijndael Algorithm"*, IEEE International Symposium on Circuits and Systems, 2004.
[7] Rainer Buchty, Nevin Heintze, and Dino Oliva, "*Cryptonite – A Programmable Crypto Processor Architecture for High-Bandwidth Applications"*, ARCS 2004, LNCS 2981, pp. 184–198, 2004.
[8] Lisa Wu, Chris Weaver, and Todd Austin, "*CryptoManiac: A Fast Flexible Architecture for Secure Communication"*, in 28th Annual International Symposium on Computer Architecture, June 2001.
[9] Ricardo Chaves, Georgi Kuzmanov, Stamatis Vassiliadis and Leonel Sousa, "*Reconfigurable Memory Based AES Co-Processor"*, 20th International Parallel and Distributed Processing Symposium, 2006.
[10] Oscar Perez, Yves Berviller, Camel Tanougast and Serge Weber, "*Comparison of various strategies of implementation of the algorithm of encryption AES on FPGA"*, IEEE International Symposium on Industrial Electronics, 2006.
[11] Yongzhi Fu, Lin Hao and Xuejie Zhang, "*Design of An Extremely High Performance Counter Mode AES Reconfigurable Processor"*, IEEE Computer Society, 2005.
[12] Alireza Hodjat, David D. Hwang, Bocheng Lai, Kris Tiri and Ingrid Verbauwhede, "*A 3.84 Gbits/s AES Crypto Coprocessor with Modes of Operation in a 0.18-µm CMOS Technology"*, GLSVLSI 2005.
[13] S. Pongyupinpanich, S. Phathumvanh, and S. Choomchuay, "*A 32 Bits Architecture For an AES System"*, International Symposium on Communications and Information Technologies, 2004.
[14] Toby Schaffer, Alan Glaser and Paul D. Franzon, "*Chip-Package Co-Implementation of a Triple DES Processor"*, IEEE Transactions on Advanced Packaging, vol. 27, no. 1, February 2004.
[15] Zhijie Shi, Xiao Yang and Ruby B. Lee, "*Arbitrary Bit Permutations in One or Two Cycles"*, IEEE 14th International Conference on Application-Specific Systems, Architectures and Processors, June 2003.
[16] Yedidya Hilewitz, Zhijie Jerry Shi and Ruby B. Lee, "*Comparing Fast Implementations of Bit Permutation Instructions"*, 38th Annual Asilomar Conference on Signals, Systems and Computers, November 2004.
[17] SystemC 2.0.1 Language Reference Manual, Open SystemC Initiative, 2003.
[18] SystemCrafter SC, Available: http://www.systemcrafter.com/