VLSI IMPLEMENTATIONS OF THE TRIPLE-DES BLOCK CIPHER

P. Kitsos, S. Goudevenos and O. Koufopavlou

VLSI Design Laboratory
Electrical and Computer Engineering Department
University of Patras, Patras, GREECE
E-mail: pkitsos@ee.upatras.gr

ABSTRACT

In this paper, VLSI implementations for the Triple-DES Block Cipher are presented. Triple-DES (TDES) is basically used in various cryptographic applications and wireless protocol security layers. Three different hardware implementations are proposed. The first two are based on the pipeline technique, while the third uses consecutive iterations for the data transformations. In addition, the used TDES S-BOXes has been implemented by both Look Up Tables (LUT) and ROM Blocks providing useful information regarding the covered area and the design throughput. The ROM approach has better performance than the LUT one but the latter is preferred in the cases that ROM blocks are not available. The proposed TDES implementations achieve high-speed performance. Especially, the throughput value for the pipeline one is equal to 7.36 Gbps.

1. INTRODUCTION

The main scope of several standards, including GPRS, EDGE, WAP and UMTS, is to meet the requirements of the wireless data communication. These standards are made for wide-area wireless data services with full mobility up to 2 Mbit/s. In addition, standards are being developed for wireless local area network multimedia communication, such as the High Performance Radio Local-Area Network, type 2 (HIPERLAN/2) [1] and ATM [2]. These standards will provide high-speed data rates, and especially HIPERLAN/2 will provide high-speed communications access to different broadband core networks and portable as well as mobile terminals. For this reason efficient cryptography algorithm implementations are required in order to supports correctly the requirements of each communication protocols.

Usually the offered security level of a communication system depends on the services and applications. For example, although the HIPERLAN/2 uses the DES block cipher for encryption, in the cases with high security level the TDES block cipher is used.

In this work, three different VLSI implementations of the Triple-DES Block Cipher are presented. The two of them are based on the pipeline technique and are suitable for high-speed application needs. The other is based on the consecutive iterations and is preferred in the case that the minimized allocated resources are the major demand. In addition, the used TDES S-BOXes have been implemented by LUTs as well as by ROM Blocks. LUTs could be used in cases where ROM blocks are not available or is intended to be used. According to our research the ROM implemented S-BOXes have better performance in comparison with the LUT approach.

Lately many designs have been proposed for the hardware implementation of the TDES [3-6]. Comparing with the implementation in [3, 4] our proposed implementation is faster while is slower comparing with the implementation in [5]. Finally, in [6] an ASIC implementation is presented. This is faster than the proposed implementations by a factor range from 0.3 to 0.4.

The paper is organized as follows: In sections 2 the TDES Block Cipher is described briefly. The proposed VLSI architectures are presented in details in section 3. The synthesis results for the FPGA implementation is shown in section 4, and the paper conclusions are given in section 5.

2. TRIPLE DES BLOCK CIPHER

The Data Encryption Standard (DES) was published by the National Bureau of Standards in 1977 [7] and reaffirmed in its final form by the Federal Information Processing Standards Publication (FIPS) in 1994 [8]. DES is a block cipher with Feistel [9] networks, which operates on data blocks of 64-bit with a key value support of 64-bit length. Triple DES is built on three DES block cipher in order to support a higher security level. It operates on the Encryption-Decryption-Encryption (EDE) mode, which uses sequentially first DES encryption, then DES decryption and last DES encryption, with the support of three different keys. The total keys length therefore is 3x64 = 192 bits. The EDE

0-7803-8163-7/03/$17.00 © 2003 IEEE

mode is illustrated in figure 1. The decryption operation of the TDES is performed such as DED mode.

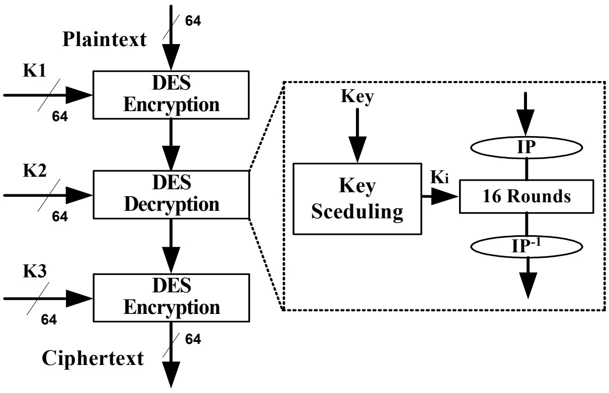

Figure 1: The Triple DES Block Cipher Architecture

3. PROPOSED TDES IMPLEMENTATION

In figure 2, the first proposed pipeline TDES architecture is presented. It consists of 48 pipeline registers between the rounds.

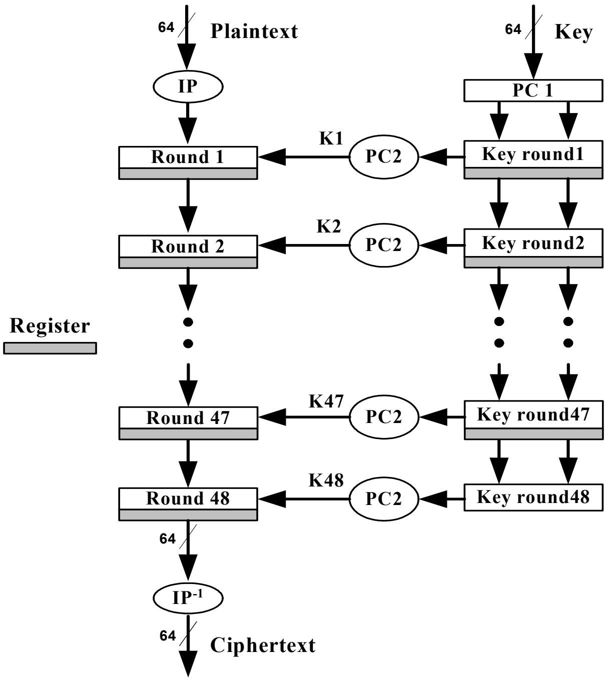

Figure 2: The 48 Stage pipeline TDES Architecture

Each DES should support encryption and decryption, because the Triple-DES algorithm in the encryption-decryption-encryption (EDE) scheme demands both. DES decryption uses the same algorithm as encryption. The only different is that the subkeys have to be generated in a reverse order comparing with the encryption. In order to produce the subkeys in reverse order they have to be cyclically shifted right (the internal on Key round$_i$ shift operations), as opposed to left for encryption.

Each DES begins with initial permutation IP and ends with the inverse initial permutation IP^{-1}. These two permutations are inverse operations. When three DES are concatenated, the initial permutation of the previous DES follows the inverse initial permutation of the current DES. There is no reason to do either permutation, since the result is no permutation at all [10]. After that, any IP-IP^{-1} pairs can be factored out of the algorithm and only the first and the last IP^{-1} need to be done. As a result, the initial permutations of the second and third DES, and the final permutations of the first and the second DES did not included of the proposed architecture. With this technique a major gain in time delay is achieved. This architecture has the possibility to process 48 independent data blocks, which drastically increases the operation throughput. The DES key scheduling can be performed on the fly. The sub-keys generated by three key scheduler units. Each, key generator consists of 16 rounds. The 64-bits input key is initially permuted, after is going trough the appropriate hardwired shifter and finally is passed through a second round permutation for each sub-key. It is implemented with pipeline stages in order to balance the pipelining in each TDES round.

A 3x64 bits memory is placed at the input to the key scheduling circuit. In this memory the three user 64 bits keys are stored in order to force the appropriate key in the appropriate time. At start the 64-bits encryption key is applied on the key scheduler. On the 17th clock cycle the decryption key is supplied on the key scheduler. Finally, on the 33rd cycle the encryption key is forced and DES operates in the encryption mode.

The second architecture consists of one DES with 16 pipeline registers between the rounds (see figure 3).

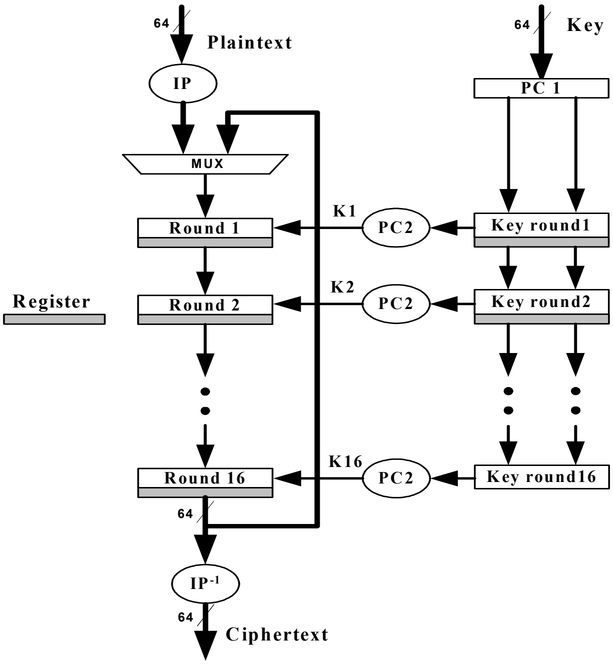

Figure 3: The 16 Stage pipeline TDES Architecture

The multiplexer determines between the new data and the output of the previous DES. This architecture has the capability of processing 16 independent data blocks with simultaneous increase the operation throughput. The DES key scheduling it can be performed on the fly. The sub-keys generation comprises 16 rounds. The sub-keys generated with the same way as referred in the 48-stage full pipeline TDES.

The above architectures are suitable for high-speed applications that support Electronic Codebook (ECB) or ATM-Counter modes of operation.

The third architecture is designed with consecutive iterations. Only one round is implemented in order to minimize the TDES implementation allocated resources. In figure 4 the TDES architecture based on the consecutive iterations is shown. In this architecture the required hardware allocated resources, comparing with the previous pipeline architectures, are reduced by a factor equal to forty eighty and sixteen respectively.

The output of the basic round is registered and one additional register is used for the input plaintext. In the multiplexer the plaintext is loaded during the initialization procedure. In the case of data transformation process the feedback permits loading new data blocks through the multiplexer.

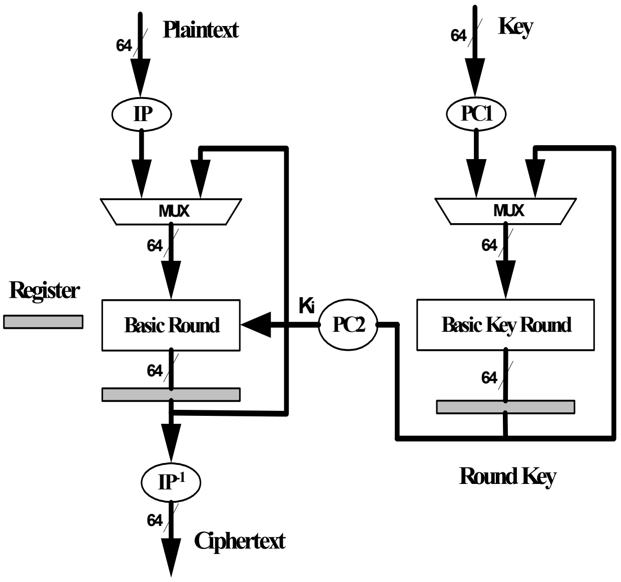

Figure 4: The Consecutive Iterations TDES Architecture

The same technique is selected for initial and final permutations as referred in the pipeline architectures. The key scheduler consists of one basic round. Consecutive round sub-keys are computed by simple rotations and permutations. The key scheduler forces the TDES by one sub key at every clock cycle. Totally, 48 clock cycles are needed in order to executed the whole TDES block cipher.

This architecture is suitable for area-restricted devices and for the Cipher Block Chaining (CBC) or Output Feedback (OFB) modes of operation.

The proposed TDES implementations could operate either with three same keys, or with two different keys. In the case that the three keys are the same the TDES has the same crypto strength as simple DES.

4. VLSI IMPLEMENTATION RESULTS

The proposed TDES architectures were captured by using VHDL. All the system components were described with structural architecture. The whole design was synthesized, placed and routed using XILINX FPGA devices [11]. Then all implementations were simulated again for the verification of the correct functionality. The TDES correct operation is validated by using the know-answer test vector that provided by [12].

For both TDES implementations the S-BOXes have been integrated by Look Up Tables (LUT) as well as by ROM Blocks.

The synthesis results for all implementations are illustrated in tables 1, 2 and 3. From these results is apparent that ROM blocks approach have better performance than this of the LUT S-BOXes one.

Table 1: The 48 Stage Pipeline TDES Implementations Synthesis Results

Implementation	LUT		ROM	
FPGA DEVICE	Xilinx V1600EBG560		Xilinx V1600EBG560	
AREA ALLOCATION	Used	Util.	Used	Util.
I/Os	326	81 %	326	81 %
Fun. Generators	28510	92 %	28380	91 %
CLB Slices	14240	91 %	14142	91 %
Dffs or Latches	10400	31 %	10400	31 %
F (MHz)	108		115	
Throughput (Gbps)	6.9		7.36	

Table 2: The 16 Stage Pipeline TDES Implementations Synthesis Results

Implementation	LUT		ROM	
FPGA DEVICE	Xilinx V400EFG676		Xilinx V400EFG676	
AREA ALLOCATION	Used	Util.	Used	Util.
I/Os	326	73 %	326	73 %
Fun. Generators	9504	99 %	9462	98.8 %
CLB Slices	4752	99 %	4715	99 %
Dffs or Latches	3472	32 %	3472	32 %
F (MHz)	108		115	
Throughput (Gbps)	2.3		2.45	

Table 3: The One-Round TDES Implementations Synthesis Results

Implementation	LUT		ROM	
FPGA DEVICE	Xilinx V200EBG352		Xilinx V200EBG352	
AREA ALLOCATION	Used	Util.	Used	Util.
I/Os	198	76 %	198	76 %
Fun. Generators	862	18 %	835	18 %
CLB Slices	431	18 %	405	17.5 %
Dffs or Latches	400	7 %	400	7 %
F (MHz)	86		91	
Throughput (Mbps)	115		121	

The comparison between the proposed TDES implementations and other previous reported is presented in Table 4. It has to be noted that the implementation of 48-stage pipeline TDES is the first, to our knowledge. For this reason only the second and third proposed designs are compared with previous ones.

Table 4: TDES Implementations Comparison

Implementation	One Round		Full Pipeline	
TDES Architecture	F (MHz)	Data rate (Mb/s)	F (MHz)	Data rate (Gb/s)
TDES in [3]	69	83	-	-
TDES in [4]	91	116	91	1.5
TDES in [5]	-	-	207	13.3
TDES in [6]	250	155	-	-
Proposed_ROM	91	121	115	2.45
Proposed_LUT	86	115	108	2.3

The proposed TDES implementations are faster comparing with the implementation in [3, 4] while is slower comparing with the implementation in [5]. In [6] the TDES is implemented in ASIC device. Finally, we have to mention that the number of the area resources in the proposed implementations is reduced significantly comparing with the implementations in [3, 4].

5. CONCLUSIONS

Three different triple DES architectures are presented in this paper. In addition, different VLSI implementations are described. The S-BOXes that used by the algorithms has been implemented either in Look Up Tables (LUT) or RAM blocks providing useful information regarding the covered area and the design throughput. The proposed designs provide high-speed performance. In addition, are more efficient in terms of area resources than many previous implementations. It is a flexible solution for any cryptographic system and security layers of wireless protocol. The proposed designs was captured entirely in VHDL language and implemented in XILINX FPGA devices. Measurement results and comparisons between the proposed and previous hardware implementations are presented.

6. REFERENCES

[1] “ETSI TS 101 761-1 V1.2.1. Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions”

[2] ATM Forum, ATM Security Speci.cation Version 1.0, ATM-SEC-01.0100, The ATM Forum, Security Working Group, 1999.

[3] Ohjun KWON, Hidenori SEIKE, Hirotsugu KAJISAKI and Takakazu KUROKAWA, “ Implementation of AES and Triple-DES cryptography using a PCI-based FPGA board”, Proc. of the International Technical Conference On Circuits/Systems, Computers and Communications 2002, ITC-CSCC-2002, Phuket, Thailand, July 16-19, 2002.

[4] Pawel Chodowiec, Kris Gaj, Peter Bellows, and Brian Schott, “Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board”, Proc. Information Security Conference, Malaga, Spain, October 1-3, 2001, pp. 220-234.

[5] Vikram Pasham and Steve Trimberger, “High Speed DES and Triple-DES Encryptor / Decryptor”, on line available in http://www.xilinx.com/xapp/xapp270.pdf

[6] Herbert Leitold, Wolfgang Mayerwieser, Udo Payer, Karl Christian Posch, Reinhard Posch, and Johannes Wolkerstorfer, “A 155 Mbps Triple-DES Network Encryptor”, Proc. Cryptographic Hardware and Embedded Systems - CHES 2000, USA, August 17-18, 2000.

[7] Data Encryption Standard, Federal Information Processing Standard (FIPS) 46, National Bureau of Standards, 1977.

[8] Federal Information Processing Standards Publication 140-1, “Security Requirements for Cryptographic Modules”, U.S. Department of Commerce/NIST, Springfield, VA: NIST, 1994.

[9] B. Schneier, “Applied Cryptography, Protocols, Algorithms, and Source Code in C”, John Wiley & Sons 1994.

[10] D. C. Feldmeier, P. R. Karn, “UNIX Password Security – Ten Years Later,” CRYPTO’89, Santa Barbara, California, USA, pp. 44-63, 1989.

[11] Xilinx, San Jose, California, USA, Virtex, 2.5 V Field Programmable Gate Arrays, 2001, www.xilinx.com.

[12] NIST Special Pubilication 800-20, "Modes of Operation Validation System for the Triple Data Encryption Algorithm", National Institute of Standard and Technology, 2000.