Comparison of AES and DES Algorithms Implemented on Virtex-6 FPGA and Microblaze Soft Core Processor

G. Renuka[1], V. Usha Shree[2], P. Chandra Sekhar Reddy[3]
[1]Department of Electronics and Communication Engineering, S R Engineering College, Warangal, Telangana, India
[2]Department of Electronics and Communication Engineering, JBREC, Hyderabad, Telangana, India
[3]Department of Electronics and Communication Engineering, JNTUH, Kukatpally, Hyderabad, Telangana, India

Article Info

Article history:

Received Apr 21, 2018
Revised Jul 21, 2018
Accepted Jul 28, 2016

Keyword:

Advanced encryption standard
Data encryption standard
FPGA
Microblaze

ABSTRACT

Encryption algorithms play a dominant role in preventing unauthorized access to important data. This paper focus on the implementations of Data Encryption Standard (DES) and Advanced Encryption Standard (AES) algorithms on Microblaze soft core Processor and also their implementations on XC6VLX240t FPGA using Verilog Hardware Description language. This paper also gives a comparison of the issues related to the hardware and software implementations of the two cryptographic algorithms.

Corresponding Author:

G. Renuka,
Departement of Electronics and Communication Engineering,
S R Engineering College (Autonomous),
Warangal, Telangana, India.
Email: renuka_g@srecwarangal.ac.in

1. INTRODUCTION

Data encryption Standard (DES) [1]-[3] is adopted in 1977 by NIST (National Institute of Standards & Technology as Federal Information Processing Standard 46 (FIPS PUB 46). DES encrypts the data of 64 bits using 56 bit key length into 64 bit cipher text. Figure 1 shows the general description of the DES algorithm for encryption. The encryption is carried out by first doing the initial permutation and then it is followed by 16 rounds consisting of permutation and substitution functions. Finally the intermediate output pass through the inverse initial permutation.

For generating the 16 48-bit keys that are used in each of the 16 rounds, the 64-bit key is used as the input to the algorithm. It is then passed to the initial permutation (PC1). The sub key is produced by circular left shifts and permutation function2 (PC2).The 64-bit plain text after initial permutation is divided into left (L) and right (R) halves of 32-bit quantities. The right half is first expanded using expander and then xored with key (Ki). It is then passed through the s-boxes and the output is then permutated and xored with the left half to produce the right half for the next round. The left half is swapped with the right half to produce the left half for the next round.

The decryption process uses the same encryption algorithm but the sub keys are applied in reverse order.In July 1998 DES has been proved insecure, when Electronic frontier Foundation (EFF) has announced that it has broken DES.The search for the more secure algorithm has given birth to a new encryption algorithm called Advanced Encryption Standard (AES) (FIPS PUB 197) [4] developed by Dr Vincent Rijmen and Dr. Joan Daemen .It is also called Rijndael algorithm named after its developers.

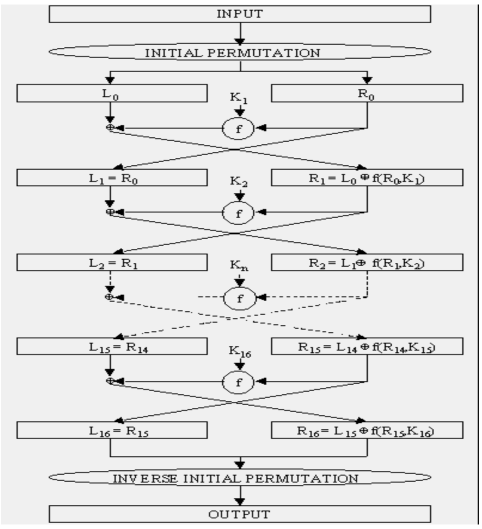

INPUT
INITIAL PERMUTATION
L_0
K_1
R_0
f
$L_1 = R_0$
K_2
$R_1 = L_0 \oplus f(R_0, K_1)$
$L_2 = R_1$
K_n
$R_2 = L_1 \oplus f(R_1, K_2)$
$L_{15} = R_{14}$
K_{16}
$R_{15} = L_{14} \oplus f(R_{14}, K_{15})$
$L_{16} = R_{15}$
$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$
INVERSE INITIAL PERMUTATION
OUTPUT

Figure 1. General structure of DES algorithm

AES uses block length of 128 bits of data and the key length be independently specified to be 128,192 and 256 bits. The algorithm consists of 10 rounds as shown in the Figure 2, for a key length of 128 bits. Unlike DES, AES does not use a feistel structure.Every round of AES comprises of 4 stages-substitute bytes, Mix columns, shift rows and add round key, except the last round which does not contain the mix column stage.

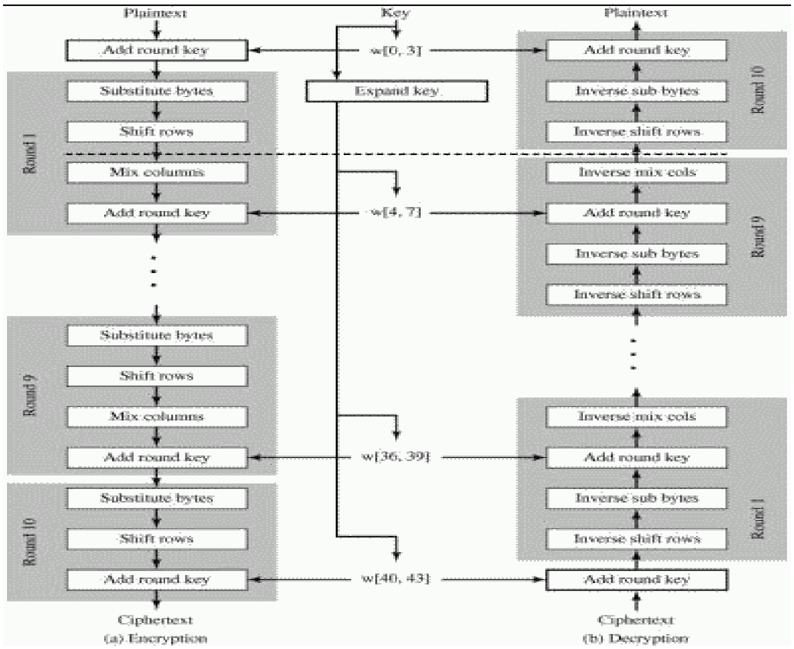


Figure 2. AES algorithm with key length of 128 bits

Initially the 128 bit key is xored with plain text in Add Round Key stage. The 128 bit data is depicted as a square matrix of bytes called as state. The 16 bytes are arranged as first four bytes as the first column, second four bytes as second column and so on. The first stage of substitute bytes uses a 16x16 matrix byte values called s-boxes. Each and every individual byte of the state is mapped into a new byte which is obtained by the intersection of row and column elements. For decryption inverse s-box is taken for substitute byte transformation. In the shift row transformation, the first row is of the state is unchanged and the second row undergoes one byte circular left shift and the third row undergoes two byte circular left shift and the fourth row undergoes a three byte circular left shifts. The decryption process uses the Inverse shift row transformation which performs circular right shift for the respective rows.The mix column transformation is carried out by performing matrix multiplication on the state. In the Add round key transformation the state is xored with the expanded key.

2. OVERVIEW OF MICROBLAZE PROCESSOR

The Microblaze is a soft core processor. It is designed for Xilinx FPGAs from Xilinx [5]. It is implemented entirely using logic fabric of Xilinx FPGAs. Microblaze is an embedded soft RISC processor which uses 32-bit instruction word, 32-bit address and data buses, 32 registers each of width 32-bit, Big-endian format with three or five pipeline stages. It uses two buses in Harvard Architecture, PLB (processor Local Bus) and LMB (Local Memory Bus). It has dedicated unidirectional point-to-point data streaming interfaces. It also supports up to 16 FSLs (Fast Simplex Links) with dedicated Cache Link ports. Figure 3 shows the microblaze core block diagram.

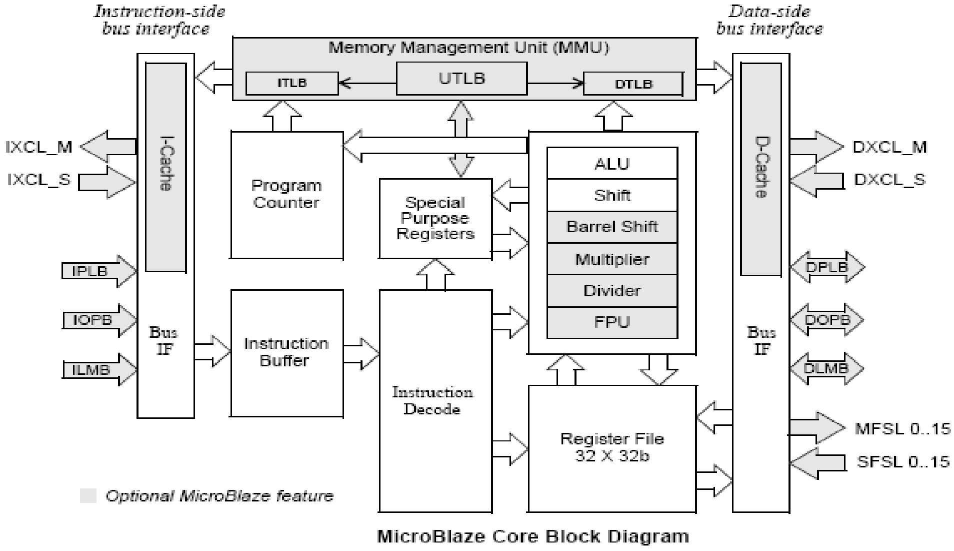


Figure 3. Microblaze core block diagram

The ALU consist of hardware multipliers/DSP48, Floating Point Unit (FPU), Barrel shifter. The FPU implements IEEE 754 single-precision, floating-point standards supporting addition, multiplication, division, comparison and subtraction. The processor includes Instruction decoder, program counter and Instruction cache.

3. XILINX PLATFORM STUDIO

The Xilinx Platform Studio (XPS), it is used for developing the hardware portion of the embedded processor system. Xilinx Embedded Development Kit (EDK) [6] is associate integrated software system tool suite.A usual embedded system [8] style project involves: hardware platform making, hardware platform verification (simulation), software system platform construction, software system application creation, and verification.

Xilinx Platform Studio package Development Kit (SDK) is in Association with Nursing integrated development atmosphere, complimentary to XPS, that's used for C/C++ embedded package application construction and verification. SDK is made on the Eclipse open source framework. Soft Development Kit (SDK) may be a suite of tools that allows you to style a package application for elite Soft IP Cores within the Xilinx Embedded Development Kit (EDK).The package application will be written during a "C or C++" then the entire embedded processor system for user application are finished, else correct & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

4. EXPERIMENTAL SETUP

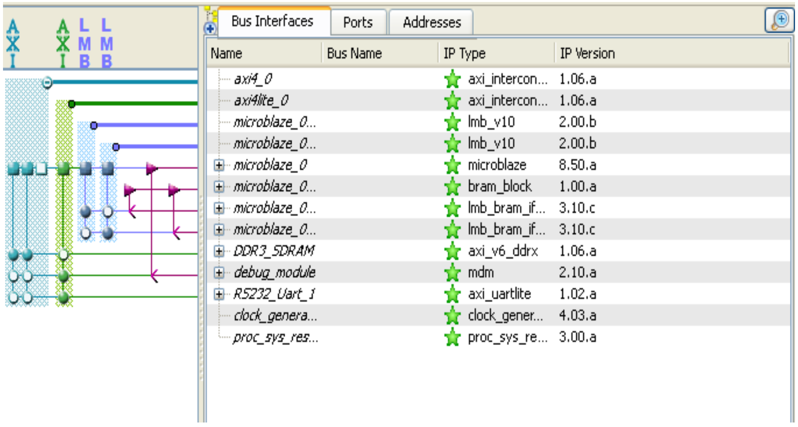
The AES and DES algorithms are described using Verilog [7]. Synthesis and Place & Route is done using Xilinx ISE 14.5. The device used for implementation is Xilinx Virtex-6 XC6VLX240t FPGA [8].

Table 1 summarizes the configuration of FPGA.

Table 1. Configuration of FPGA

Family	Virtex-6
Device	XC6VLX240T
Package	FF1156
Speed grade	-1

There are many soft core processors available like PowerPC [9], NIOS3 [10], LEON3 [11] and Microblaze [12], [13]. We have chosen Xilinx Microblaze for our implementations. The designs are created with Xilinx Platform Studio (XPS) as shown in the Figure 4.



Name	Bus Name	IP Type	IP Version
axi4_0		axi_intercon...	1.06.a
axi4lite_0		axi_intercon...	1.06.a
microblaze_0...		lmb_v10	2.00.b
microblaze_0...		lmb_v10	2.00.b
⊞ microblaze_0		microblaze	8.50.a
⊞ microblaze_0...		bram_block	1.00.a
⊞ microblaze_0...		lmb_bram_if...	3.10.c
⊞ microblaze_0...		lmb_bram_if...	3.10.c
⊞ DDR3_SDRAM		axi_v6_ddrx	1.06.a
⊞ debug_module		mdm	2.10.a
⊞ RS232_Uart_1		axi_uartlite	1.02.a
clock_genera...		clock_gener...	4.03.a
proc_sys_res...		proc_sys_re...	3.00.a

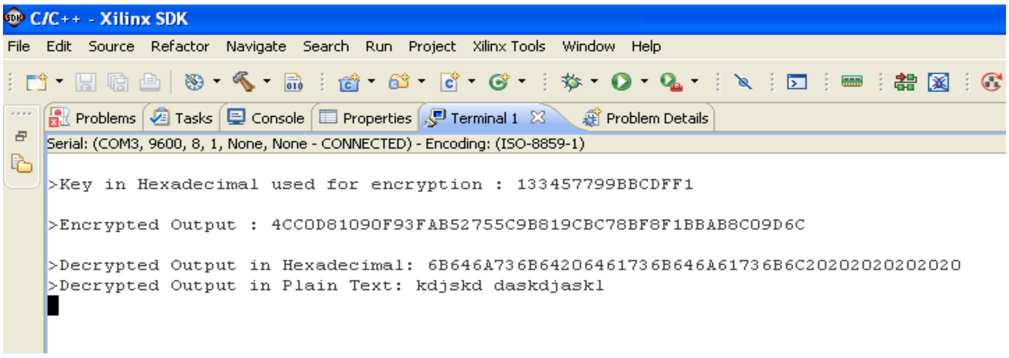
Figure 4. System assembly view for soft core processor

The programs for AES and DES are written using C code. They are compiled in SDK (Software Development Kit) [14]. The SDK provides a platform for applications targeted for embedded Microblaze.

5. RESULTS AND ANALYSIS

5.1. Performance of DES algorithm

The results of the DES algorithms implemented on Xilinx Microblaze are shown in Figure 5 and the simulation results of DES are shown in Figure 6.



```
Serial: (COM3, 9600, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)

>Key in Hexadecimal used for encryption : 133457799BBCDFF1

>Encrypted Output : 4CC0D81090F93FAB52755C9B819CBC78BF8F1BBAB8C09D6C

>Decrypted Output in Hexadecimal: 6B646A736B6420646173 6B646A61736B6C202020202020 20
>Decrypted Output in Plain Text: kdjskd daskdjaskl
```

Figure 5. Results of DES output seen through HyperTerminal

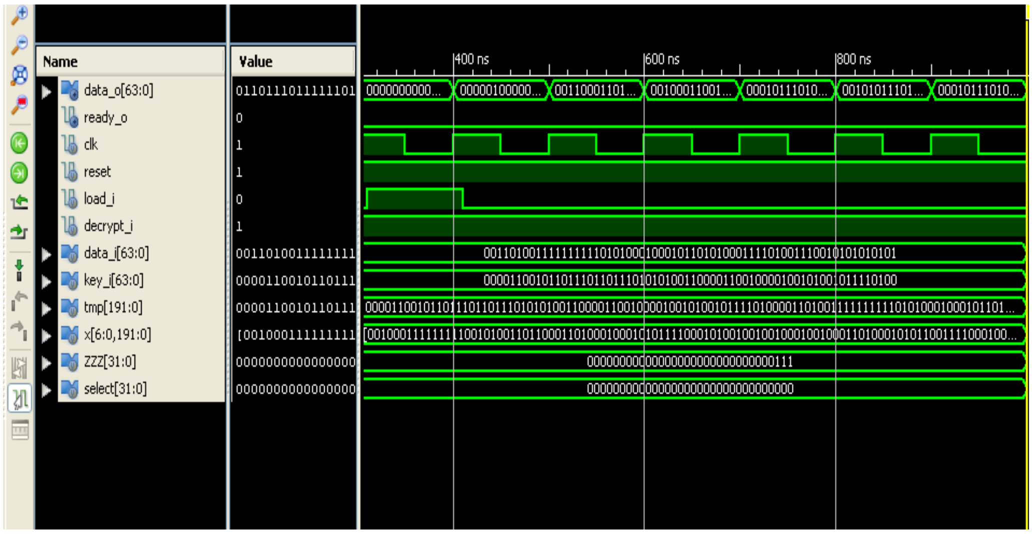
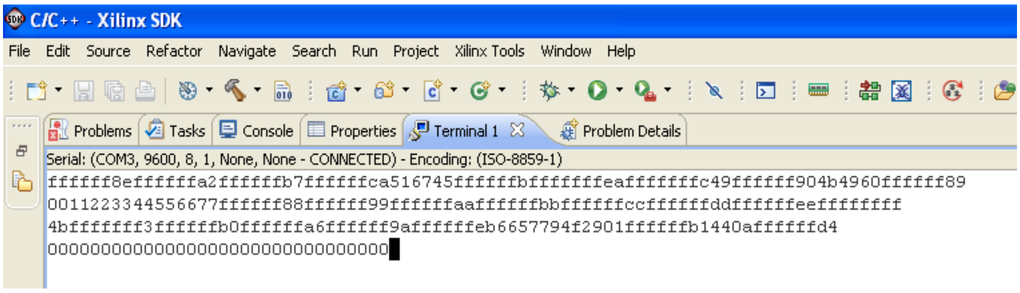


Figure 6. Simulation of 128 bit DES encryption

5.2. Performance of AES algorithm

The results of the AES algorithms implemented on Xilinx Microblaze are shown in Figure 7 and the Simulation results of AES are in Figure 8.


C/C++ - Xilinx SDK
File Edit Source Refactor Navigate Search Run Project Xilinx Tools Window Help
Problems
Tasks
Console
Properties
Terminal 1
Problem Details

```
Serial: (COM3, 9600, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)
ffffff8effffffa2ffffffb7ffffffca516745ffffffbffffffeafffffffc49ffffff904b4960ffffff89
00112233445566 77ffffff88ffffff99ffffffaaffffffbbffffffccffffffddffffffeeffffffff
4bfffffff3ffffffb0ffffffa6ffffff9affffffeb6657794f2901ffffffb1440afffffd4
00000000000000000000000000000000
```

Figure 7. Results of AES output seen through HyperTerminal

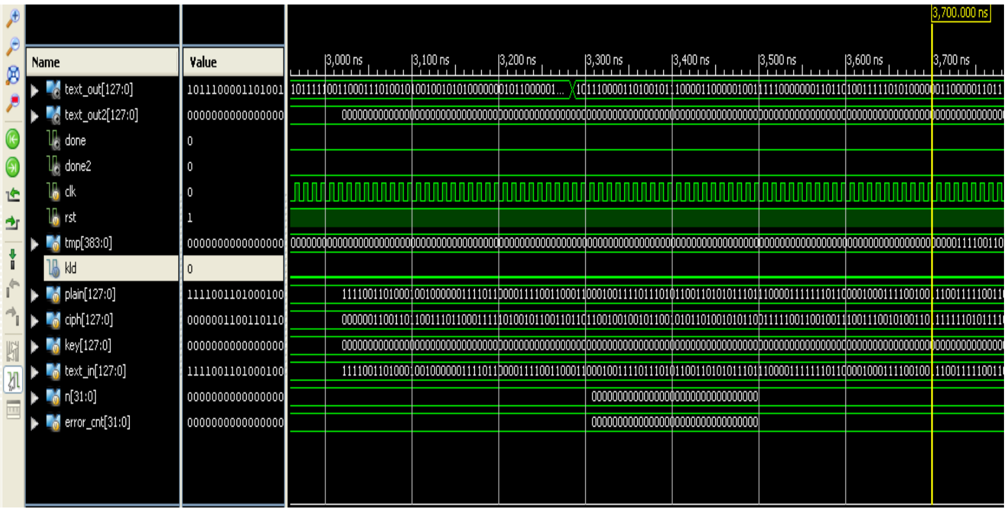


Figure 8. Simulation of 128 bit AES encryption

The Table 2 summerises the implementation results of AES and DES algorithms.

Table 2. Comparision of AES and DES Algorithms

	No of slices used	Max. freq(MHz)
AES algorithm using verilog	1456	181.752
DES algorithm using verilog	450	264.896
AES algorithm using microblaze	12,611	100.664

6. CONCLUSION

Although DES is replaced by AES, the study of DES algorithm provides us with an insight in understanding the principles used in symmetric ciphers. A comparison of both implementations reveals that a direct Verilog implementation gives a better speed and area option rather then implementing in the soft core Microblaze processor. But the design effort to implement the algorithms on Microblaze is comparatively less than the direct Verilog implementation. Therefore, a tradeoff has to be made between the design effort, area and speed optimizations when choosing the options for various implementations.

REFERENCES

[1] "Data encryption standard (DES)", National Bureau of Standards (U. S.), Federal Information Processing Standards Publication 46, National Technical Information Service,Springfield, VA, Apr. 1977.

[2] W. Stallings, "Cryptography and Network Security: Principles and Practice", 4th ed, Prentice-Hall, 2006

[3] Baker, W, "Introduction to the Analysis of the Data Encryption Standard (DES) Laguna Hills", CA: Aegean Park Presss, 1991.

[4] Daemen, J and Rijmen, V, A Specification for The AES Algorithm.NIST (National Institute of Standardsand Technology)http://csrc.nist.gov/archive/aes/rijndael/wsdindex.html, 2010.

[5] www.xilinx.com/

[6] Xilinx: Getting Started with the Virtex-6 FPGA ML605 Embedded KitUG730 (v1.1) June 14, 2010

[7] Samir Palnitkar, "Verilog HDL, A Guide to Digital Design and Synthesis", Prentice Hall, 2003

[8] Xilinx: Embedded System Tools Reference Guide UG111 September 16, 2009

[9] Xilinx: ML605 Hardware User Guide UG534 (v1.8) October 2, 2012

[10] "IBM PowerPC Quick Reference Guide", IBM Corp. 2005.

[11] "NIOS 3.0 CPU Data Sheet", Altera Corporation, 2004 http:Hwww.altera.com/literature/ds/ds_nios_cpu.pdf

[12] Oukili, Soufiane, and Seddik Bri, "High throughput FPGA Implementation of Data Encryption Standard with time variable sub-keys", *International Journal of Electrical and Computer* Engineering, vol. 6, no. 1, p. 298, 2016.

[13] Espalmado, J. M. and Arboleda, E., "DARE Algorithm: A New Security Protocol by Integration of different Cryptographic Techniques", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 2, pp. 1032-1041, 2017.

[14] Mone, Shubhada Parashar, and Sunita S. Dhotre, "Enforcing multi-user security policies in cloud computing", *International Journal of Electrical and Computer Engineering*, vol. 3, no. 4, p. 504, 2013.