www.ijsetr.com

International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.05,Issue.03,
February-2016,
Pages:0407-0411

Design of 128-Bit AES Algorithm with UART Soft Core for Data Secure

VISHAL NANDOKAR[1], PROF. HARSHVARDHAN UPADHYAY[2]

[1]PG Scholar, Dept of VLSI & ES, Shree Ramchandra College of Engineering, Lonikand, Pune, India.
[2]Assistant Professor, Shree Ramchandra College of Engineering, Lonikand, Pune, India.

Abstract: There are many encryption algorithms, but Advanced Encryption Standard (AES) 128-bit to be used in many applications for protecting sensitive information/data. Typical examples are high-throughput design and low-area design. The former AES algorithm emphasizes its throughput using pipeline pattern. Its biggest advantage is to improve the system throughput, but there is a clear disadvantage that is at the cost of on -chip resources. Generally, the embedded applications do not require very fast speed but have a very limited area and low power consumption. Due to the importance of the AES algorithm and the numerous applications that it has, our main concern AES encryption is based on an iterative round-looping architecture. Which provide lowest area. Physical design of a circuit is the phase that precedes the fabrication of a circuit. The performance of the circuit, its area, its yield, its reliability depends on the layout of the circuit. In this paper we propose TAP (Time, Area, and Power) Analysis on UART implementation with AES algorithm for secure Data Transmission and Reception. The complete design is described in Hardware Description Language (HDL) and is functionally verified using XilinxISE 14.6 software for Synthesis, ModelSim 6.4 for Simulation.

Keywords: AES, Encryption, Decryption, Field Programmable Gate Array (FPGA), UART.

I. INTRODUCTION

Data security is an essential objective for the military and diplomatic services which have many commercial uses and applications such as electronic banking, electronic mail, internet network service, messaging networks etc. As an efficient and cost-effective cryptographic algorithm AES[1] algorithm has broad applications, including smart cards and cellular phones, WWW servers and automated teller machines (ATMs). Establishing reliable communication between multiple FPGA systems/cards is an essential component for developing complex real time systems used for applications like real time data acquisition and processing. But there exist no implementation based on real time data encryption and decryption over a communication interface between multiple FPGA systems using an RTOS, where each thread is capable of running the encryption and decryption individually. The AES algorithm has an SPN (Substitution Permutation Network) structure. Because of the growing requirements for high speed secure communications, the application of AES algorithm in UART (Universal Asynchronous Receiver Transmitter) module which is a widely used in serial data communication to support full duplex serial communication is proposed here. The UART is an integrated circuit which handles the conversion between serial and parallel data [7-9].

In this proposed design each bytes of data per clock cycle is stored in a 128-bit shift register and then encrypted using AES encryption algorithm after that each encrypted bytes are shifted serially to the input of theUART transmitter. The UART transmitter takes bytes of data and transmits the individual bits in a sequential way. The reverse operation is performed in the UART receiver portion. The UART receiver re-assembles the bits into complete byte. These received bytes are serially stored in a 128-bit shift register for the AES decryption operation. The proposed UART module is operated without parity bit, eight data bit and one stop bit. Here we choose AES-128 algorithm for encryption and decryption operation. The round transformations of AES-128 for both encryption and decryption are simulated using an iterative design approach in order to minimize the area. AES Algorithm: The AES algorithm is an iterative algorithm composed of 10 rounds. After the initial secret key addition (roundkey (0)), the first 9 rounds are identical, with different the final round [10]. Each of the first 9 rounds consists of 4 transformations: Sub Bytes, Shift Rows, Mix-Columns and AddRoundKey. The final round excludes the Mix Columns transformation. The above encryption scheme can be inverted to get a decryption structure. The Sub Bytes transformation is a non-linear byte substitution that operates independently on each byte of the State using a substitution table (S-box).

This S-box is constructed by composing two transformations multiplicative inverse in the finite field GF(28) and affine transformation. UART: A Universal Asynchronous Receiver Transmitter (UART) is a circuit that sends parallel data through a serial line. A UART includes a transmitter and a receiver. So, the main function of a UART is the conversion of parallel-to- serial when transmitting and serial- to parallel when receiving.The transmitter is essentially a special shift register that loads data in parallel and then shifts it out bit by bit at a specific rate.The receiver, on the other hand, shifts in data bit by bit and then reassembles the data. The serial line is 1 when it is idle. The transmission starts with a start bit, which is 0, followed by data bits and an optional parity bit, and ends with stop bits which are 1. Our proposed work is an FPGA based design and implementation of the AES-128 algorithm on Real Time Operating System.We have successfully established a secured link between two FPGA systems through RS232 link and have achieved good throughput with a minimum number of resource utilization. The total system functions like a complete system where data are taken from key board in real time by the FPGA, the encryption is done using a thread running on the RTOS. Through the RS232 communication link the cipher data is being transmitted to another FPGA, where the decryption is done by another thread of the RTOS. The decrypted data is visualized on the HyperTerminal for verification.

II. AES ALGORITHM

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plain-text.

A. AES Encryption

The AES algorithm operates on a 128-bit block of data and executed Nr - 1 loop times. A loop is called a round and the number of iterations of a loop, Nr, can be 10, 12, or 14 depending on the key length. The key length is 128, 192 or 256 bits in length respectively. The first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and no MixCoulmns transformation is performed in the last round. In this paper, we use the key length of 128 bits (AES-128) as a model for general explanation. An outline of AES encryption is given in Fig. 1.

Sub Bytes Transformation: The Sub Bytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The Sub Bytes transformation is done using a once-pre calculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. More details of the method of calculating the S-box table refers to [4]. In this design, we use a look-up table as shown in Table I. This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation. This approach avoids complexity of hardware implementation and has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency.

Shift Rows Transformation: In ShiftRows transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

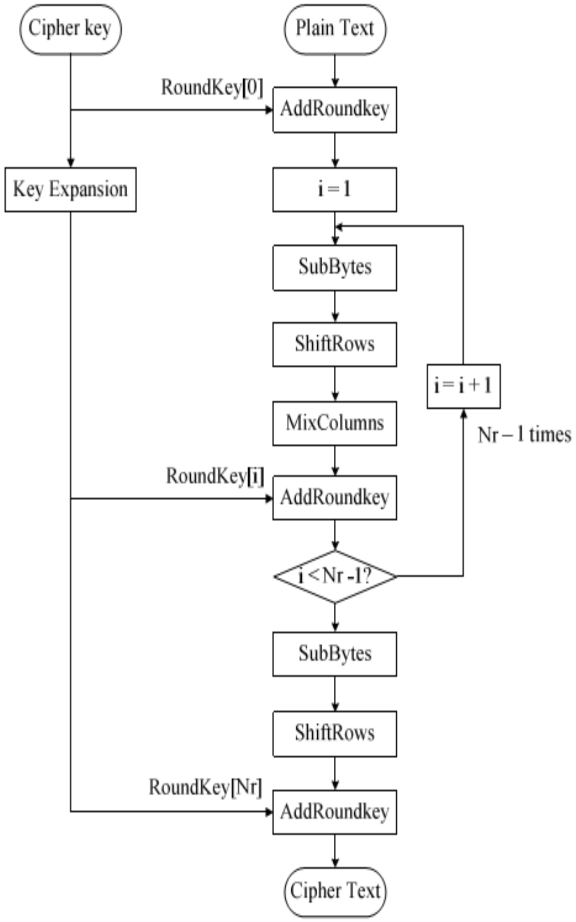

Fig.1. AES encryption structure.

Mix Columns Transformation: In Mix Columns transformation, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4+ 1 with a fixed polynomial c(x), given by: c(x)={03}x3+ {01}x2+ {01}x + {02}.

AddRoundKey Transformation: In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation. The RoundKey of each round is derived from the main key using the KeyExpansion algorithm. The encryption/decryption algorithm needs eleven 128-bit RoundKey, which are denoted RoundKey[0] RoundKey[10](the first RoundKey [0] is the main key).

B. AES Decryption

Decryption is a reverse of encryption which inverse round transformations to computes out the original plaintext of an encrypted cipher-text in reverse order. The round transformation of decryption uses the functions AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes successively.

AddRoundKey: AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order. The description of the other transformations will be given as follows.

InvShiftRowsTransformation: InvShiftRows exactly functions the same as ShiftRows,only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

InvSubBytesTransformation: The InvSubBytes transformation is done using a once-precalculated substitution table called InvS-box. That Inv S-box table contains 256 numbers (from 0 to 255) and their corresponding values. InvS-box is presented in Table II.

InvMixColumns Transformation: In the InvMixColumns transformation, the polynomials of degree less than 4 over GF(28), which coefficients are the elements in the columns of the state, are multiplied modulo (x4+ 1) by a fixed polynomial d(x) = {0B}x3+ {0D}x2+ {09}x + {0E}, where {0B}, {0D}; {09}, {0E} denote hexadecimal values.

III. FPGA BASED ARCHITECTURE

A detailed description of the proposed combined architecture for UART module with AES crypto circuit is explained in this section. The design consists of five main units; the first unit is Encryption Function unit which loads the eight bit words per clock cycle in 128 bit serial in parallel out shift register and then encrypt the data using AES-128 encryption algorithm, after that serially transmit the eight bit encrypted word per clock cycle to the input of the UART_Tx module through 128 bit parallel in 8 bit serial out shift register, the second unit is UART_Tx module that frames the eight bit word coming from AES-128 encryption unit with a START bit (logic '0') at the beginning, and a STOP bit (logic'l') at the end of the word and sends the framing information in a serial manner from the Least Significant Bit(LSB) to the Most Significant Bit (MSB). The third main module is UART_Rx module which performs the reverse operation of UART_Tx module. It removes the START bit and STOP bit and collects the data in its output port as eight bit word format. Decryption Function unit, the forth unit performs the decryption operation and sends the decrypted data to the output of the UART module. Fifth unit is clock_generator module which takes two main signals (system clock and reset) as its input and generates the new clock signals. All the five basic units of the proposed architecture are shown in Fig2. All modules are designed using Hardware Description Language (HDL).

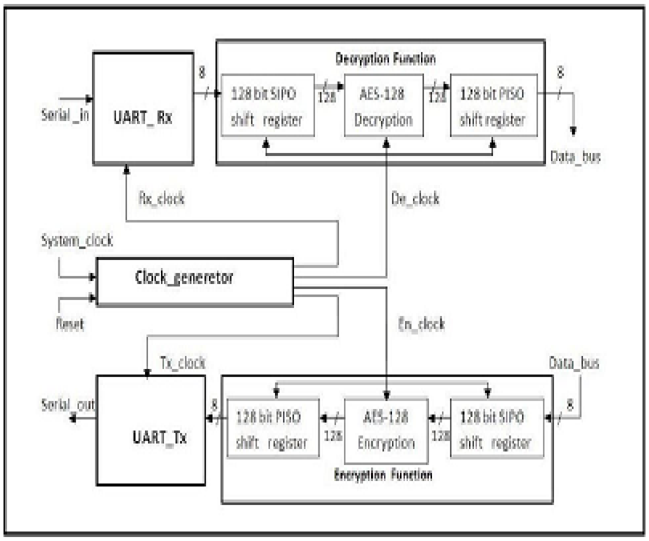

Fig 2. Proposed UART architecture.

A. Encryption Function Module

Encryption Function Module consists of three sub units, two 128-bit shift register and AES encryption module. The first serial in parallel out shift register stores the data bytes in each clock cycle and sends them to the AES-128 Encryption module for encryption. The 128 bit encrypted data is transmitted to the UART transmitter through parallel in serial out shift register. This AES Encryption module perform the encryption operation on 128bit of data using AES algorithm which is a symmetric block cipher that processes data blocks of 128 bits using three different cipher key lengths 128, 192 and 256 bits. The proposed design supports the AES-128 Encryption. In encryption procedure after an initial round key addition, a round function consisting of four different transformations – Sub Bytes, Shift Rows, Mix Columns and Add Round Keys are applied to the 128 bit data block as shown in the Fig. 3. This single round function is performed iteratively 10 times. In the last round of encryption the round function does not contain the Mix Column transformation.

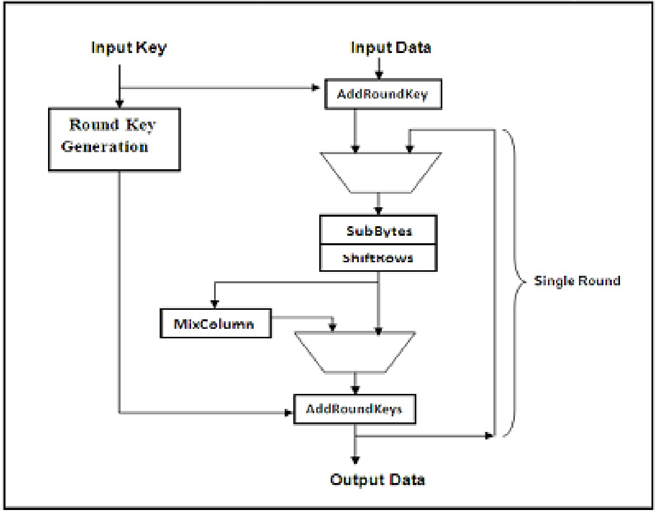

Fig.3. Iterative Architecture of AES Encryption.

B. UART Transmitter Module

UART transmitter module ie UART_Tx that frames the eight bit word coming from AES-128 encryption unit with a START bit (logic '0') at the beginning, and a STOP bit (logic 'l') at the end of the word and sends the framing information in a serial manner from the Least Significant Bit(LSB) to the Most Significant Bit(MSB). The architecture of the transmitter will consist of a controller, a data register (XMT_datareg), a data shift register(XMT_shftreg) and a status register(bit_count) to count the bits that are transmitted. Load_XMT_datareg signal is asserted to indicate that XMT_datareg now contains the data_bus value and that is now transferred to the internal shift register that is XMT_shftreg shown in the above Fig. 4.

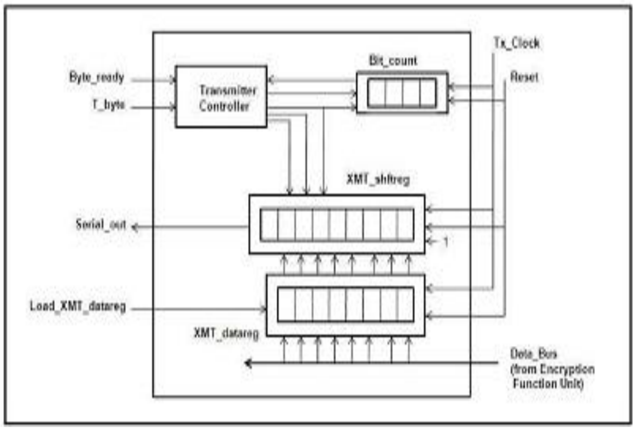

Fig.4. Block diagram of UART Transmitter.

C. Decryption Function Module

Decryption Function Module consists of three sub units, two 128 bit shift register and one AES decryption module. It performs the operation in the same manner like encryption function module, instead the AES-128 encryption operation it performed the AES-128 decryption operation on the received data from UART_Rx module. The individual transformations performed in the decryption process are Inv Sub Bytes, Inv Shift Rows, Inv Mix Columns and Add Round keys shown in the Fig5. These transformations are the inverse of the corresponding transformations in the encryption process.

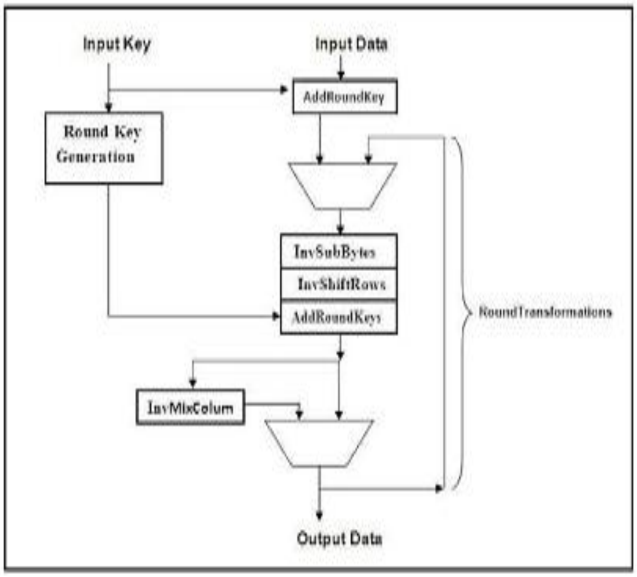

Fig 5. Iterative Architecture of AES Decryption.

D. UART Receiver Module

The UART receiver module that is UART_Rx performed the task of receiving the serial bit stream of data, removing the start bit and transferring the data in a parallel format to a storage register connected to the host data bus. The data arrives at a standard bit rate, but it is not necessarily synchronized with the internal clock at the host of the receiver, and the transmitter's clock is not available to the receiver. This issue of synchronization is resolved by generating a local clock (Rx_clock) at a higher frequency and using it to sample the received data in a manner that preserves the integrity of the data. The UART receiver circuit includes RCV_shftreg (the shift register receiving Serial_in), RCV_datareg (the 8-bit register holding the received word) and Receiver controller to control the UART receiver circuit as shown in the Fig. 6.

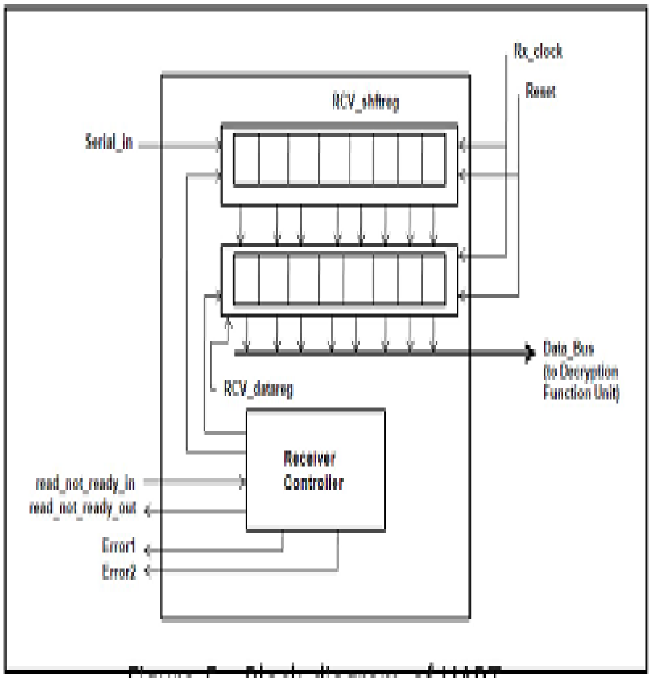

Fig 6. Block diagram of UART Receiver.

E. Clock Generator Module

The rate at which the data is transmitted is known as baud rate. UART receiver module(UART_Rx) operates on the frequency which is 8 times higher than transmitter. Clock generator module provides the four clock signals for UART transmitter, UART receiver, and encryption and decryption unit to maintain data integrity between the four different modules. In our design clock generator module uses 100MHz clock frequency. We chose 12500000 baud rate for our design.

IV. EXPERIMENTAL RESULTS

The proposed UART design and its sub modules are implemented using Hardware Description Language in Xilinx ISE 14.6. To minimize the hardware implementation iterative looping technique is used for AES encryption and decryption algorithm and results as shown in Figs.7 to 9.

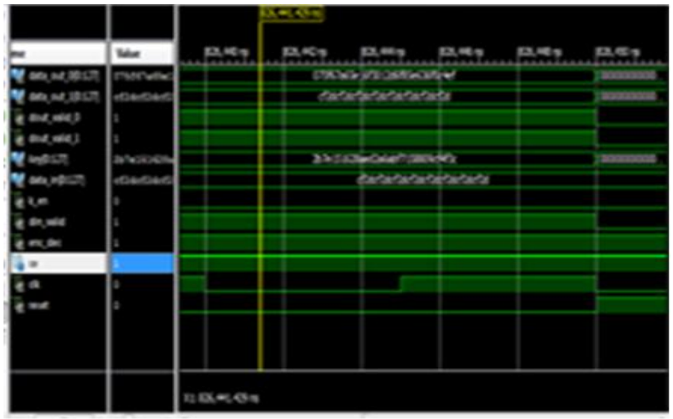

Fig.7. Timing simulation of AES encryption algorithm.

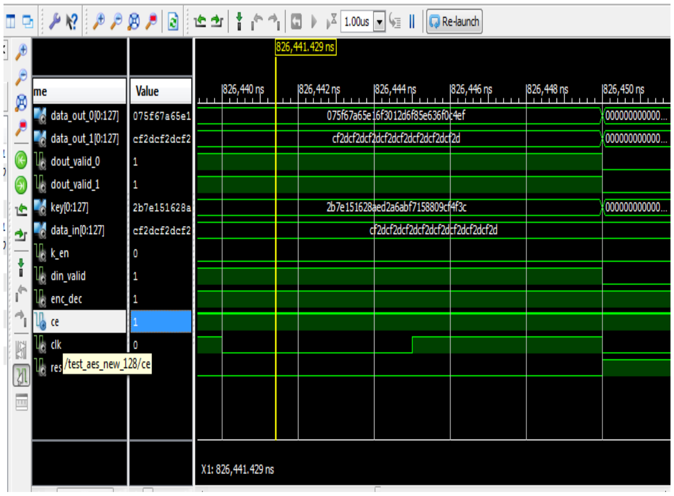

Fig.8. Timing simulation of AES decryption algorithm.

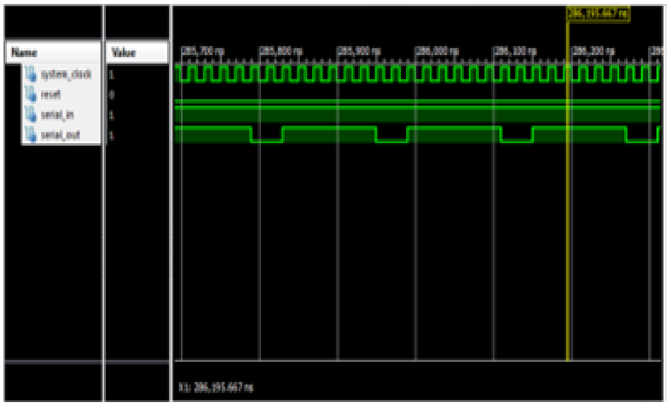

Fig.9. Timing simulation of UART Output.

V. CONCLUSION

This paper provides a detailed description of a hardware implementation for proposed multi-rate encryption unit using FPGA technology. It goes into more details in the design of the building blocks of the architecture of this unit. And also provide the details of the simulation results and hardware testing. All the main system components are described in VHDL. All interfaces and interconnect modules used for interfacing, processing and controlling are described in VHDL.

VI. REFERENCES

[1] N.Hamdy, K.Shehata, H.El-demerdash. “Design and Implementation of Encryption Unit Based on Customized AES Algorithm” IJVIPNS International Journal of Video & Image Processing and Network Security, Vol: 11 No: 01, February 2011.

[2] National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), Federal Information Processing Standards Publications (FIPS) PUBS #197, 2001.

[3] J. Daemen, V. Rijmen, AES proposal: Rijndael Document version 2, 1999.

[4]Douglas L. Perry, "VHDL: Programming by Example", Fourth Edition, 2002.

[5] Volnei A. Pedroni, “Circuit Design with VHDL” Fourth Edition, 2004.

[6] William Stallings, "Cryptography and Network Security Principles and Practices", Fourth Edition, 2005.

[7] Clive “Max” Maxfield, “The Design Warrior’s Guide to FPGAs” 2004.

[8] Spartan-3A/3AN FPGA Starter Kit Board User Guide UG334 (v1.1) June 19, 2008.

[9] Ben Cohen, “VHDL Coding Styles And Methodologies”, Kluwer Academic publishers, United States, 1995.