Biohouse, Volume 3: Issue 2, Mar-Apr 2017

Biohouse Journal of Research in Engineering

ISSN 2379-1535

Triple Data Encryption and Decryption

Dr. P.V.Rao

Prof. and R&D Head, Dept Electronics and communication Engineering,
Rajarajeswari College of Engineering,
Bangalore, India

Abstract: *TDES (Triple Data Encryption Standard) is an enhanced Data Encryption Standard Algorithm, which uses a longer key length. The DES algorithm is less efficient and AES (advance Encryption Standard) algorithm consumes more area in a chip. The longer key length used in Triple DES reduces the attacks, it takes more time for the hacker to break TDES as compared to DES and improves the reliability. The Triple DES procedure is the same as regular DES, repeats three times. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key. In this work, the design of Cryptography Algorithm DES which contains the two processes encryption and decryption and both work on the same algorithm. Key is placed from 1 to 16 in encryption and 16 to 1 in decryption. The encipher (converting plain text to cipher text) and decipher. These two processes can be executed efficiently by means of multiplex-based design. The efficient key implementation used in this design reduces the hardware area by 50%,, compared to the conventional approach. TDES functionality is verified for both multiplexer based and conventional based designs with the necessary test vectors for input data, 3keys, and Decrypt/Encrypt signals. In this work, both the designs are synthesized using TSMC 130nm technology. Multiplexer based design has an area of 5,53,311.6mm2 and power of 1.196 mW as compared to the conventional design which has an area of 1,33,8798mm2 and power of 2.363mW. These results show that adopted multiplexer based design excels the conventional design by 50% in terms of power and area in RTL level.*

Triple Data Encryption Standard (DES) processor is fast data encryption is becoming a more important requirement for applications such as secure networking. Using DES should not give high security data, so data can be encrypted three times with the same algorithm to realize Triple DES. Through the utilization of dynamic, a fast realization can be achieved. The synthesis of dynamic logic is difficult, because there are no synthesis tools, which support such logic styles. In this paper, the main focus is on the design flow of the DES chip. It discusses the basics for encryption and dynamic logic and explains the performance goals and the resulting chip structure briefly. Further, the single steps for the design of the processor are introduced trying to keep the relationship between the synthesis and the verification. The performance comparison is made with the results obtained .

The DES algorithm is a re-circulating, 64-bit, block product cipher whose security is based on a secret key. The DES keys are 64-bits binary vectors consisting of 56 information bits and 8 parity bits. The parity bits are reserved for error detection purposes and are not used by the encryption algorithm. The 56 information bits are used by the enciphering and deciphering operations and are referred to as the active key.

In the enciphering computation, a block to be enciphered is subjected to an Initial Permutation (IP), then to a complex key dependent computation and finally to a permutation which is the inverse of the IP. The key-dependent computation can be defined in terms-1 of a function f, called the cipher function, and a function KS, called the key schedule.

The Function (F) involves E-permutation operators, Substitution tables (S-boxes), and Permutations (P). The 64 bits input block is divided in to two halves, each consisting of 32 bits. One half is used as input to the Function F, and the result is exclusive-OR ed to the other half. After one iteration, or round, the two halves of data are swapped, and the operation is performed again. The DES algorithm uses 16 rounds to produce a re-circulating block product cipher. The cipher produced by the algorithm displays no correlation to the input. Every bit of the output depends on every bit of the input and on every bit of the active key.

For a thorough discussion of the DES algorithm and its components, consult FIPS PUB 46-3. Guidelines on the proper usage of the DES are published in FIPS PUB 74, Guidelines for Implementing and Using the NBS Data Encryption Standard. The non-linear substitution tables, or S-boxes, constitute an important part of the algorithm. The purpose of the S-boxes is to ensure that the algorithm is not linear. There are eight different S-boxes. Figure 1 displays one of these. Each S-box contains 64 entries, organized as a 4 × 16 matrix. Each entry is a four bits binary number, represented as 0-15. A particular entry in a single S-box is selected in six bits (i.e., two are select a row and four select a column).

The entry in the corresponding row and column is the output for that input. Each row in each S-box is a permutation of the numbers 0-15, so no entry is repeated in any one row. The output of the parallel connection of eight S-boxes is 32 bits. The role of the Permutation P is to thoroughly mix the data bits so they cannot be traced back through the S-boxes. The initial and Final Permutations are byte oriented, and the data is output eight bits at a time. The operator E expands a 32 bits input to a 48 bits output that is added mod two to the round key.

The permutations in the key-schedule, PC1 and PC2, intermix the bits that result from the S-box substitution in a complex way to prevent bit tracing. Each permutation is a linear operator, and so can be thought of as an n x m matrix and can be validated completely if it operates correctly on an appropriate maximal linearly independent set of input vectors, i.e., a suitable basis.

The Data Encryption Standard (DES) shall consist of the following Data Encryption Algorithm (DES) and Triple Data Encryption Algorithm (TDEA, as described in ANSI X9.52).

These devices shall be designed in such a way that they may be used in a computer system or network to provide cryptographic protection to binary coded data. The method of implementation will depend on the application and environment. The devices shall be implemented in such a way that they may be tested and validated as accurately performing the transformations specified in the following algorithms.

In this recommendation, each TDEA shows in Fig. 1 forward and inverse cipher operation is a compound operation of DEA forward and inverse transformations as specified.

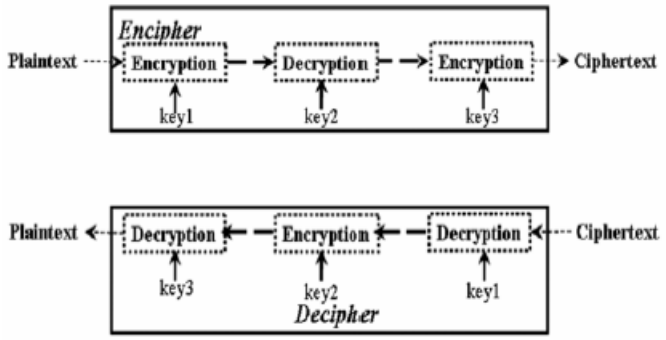

Figure 1: Block Diagram of TDEA [1]

A TDEA key consists of three keys for the cryptographic engine (Key1, Key2 and Key3); the three keys are also referred to as a key bundle (KEY). Two options for the selection of the keys in a key bundle are allowed. Option 1, the preferred option, employs three mutually independent keys (i.e. Key1, Key2 and Key3, where Key1≠ Key2≠ Key3≠ Key1). Option 2 employs two mutually independent keys and a third key that is the same as the first key (i.e. Key1, Key2 and Key3, where Key1≠ Key2 and Key3 = Key1). A key bundle shall not consist of three identical keys [2].

DEA forward and inverse transformations on data using key bundle KEY. The following operations are used:

• TDEA forward cipher operation: the transformation of a 64-bits block d into a 64-bits block O that is defined as follows:

$$O = F_{key3}(I_{key2}(F_{key1}(d))) \qquad (1)$$

TDEA inverse cipher operation: the transformation of a 64-bits block d into a 64- bits block O that is defined as follows:

$$O = I_{key3}(F_{key2}(I_{key1}(d))) \qquad (2)$$

This recommendation specifies the following keying options for a TDEA key bundle (Key1, Key2, and Key3)

• Keying Option 1: Key1, Key2 and Key3 are independent keys (i.e., Key1≠ Key2, Key3≠ Key1)

• Keying Option 2: K1 and K2 are independent keys (i.e., Key1≠ Key2), and Key3 = Key1

The TDEA keys shall be managed in accordance with NIST Special Publication (SP) 800-57, Recommendation for Key Managements. SP 800-57 also specifies time frames during which the TDEA keying options may be used. The following specifications for keys shall be met in implementing the TDEA modes of operation [3].

For all TDEA modes of operation, three cryptographic keys (Key1, Key2, and Key3) define a TDEA key bundle. The bundle and the individual keys must:

• Be secret
• Be generated randomly or pseudo randomly
• Be independent of other key bundles
• Have integrity whereby each key in the bundle has not been altered in an unauthorized manner since the time it was generated, transmitted, or stored by an authorized source
• Be used in the appropriate order as specified by the particular mode
• Be considered a fixed quantity in which an individual key cannot be manipulated while leaving the other two keys unchanged and cannot be unbundled except for its designated purpose [4]

The Encryption and Decryption algorithm is implemented in HDL coding. DES and TDES architecture is designed using key scheduling algorithm to reduce the area and power. TDES functionality is verified for both multiplexes based and conventional based designs with necessary the test vector (Input data, 3keys, and Decrypt/Encrypt signals). Both the designs are synthesized using TSMC 130nm technology Multiplexed based design has area 553311.6mm2 power is 1.196mW and conventional design has 1338798mm2 and power 2.363mW In this work, the design results shows that adopted Multiplexed based excelled conventional design by 50% in terms of power and area.

REFERENCES

[1] Jaramillo-Villegas, Jose A, Correa-Agudelo, Esteban M, Gomez-Londono and Rene, "TDES Implementation in a Reconfigurable Computing Enviroment", 4th Southern Conference on Programmable Logic, 2008, 26 Issue 28 pp 191-195, March 2008.

[2] Arich.T, and Eleuldj. M, "Hardware Implementations of the Data Encryption Standard", 14th International Conference on Microelectronics (ICM), 2002, pp. 100- 103, December 2002

[3] Mostafa-Sami M. Mostafa, Safia. H.D eif and Hisham.Abd Elazeem. Ismail. Kholidy, "ULTRA GRIDSEC Peer-to-Peer Computational Grid Middleware Security Using High Performance Symmetric Key Cryptography" 4th Southern Conference on Programmable Logic, 26 Issue 28, pp. 137-142, April 2008.

[4] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High Performance Microprocessor Design", Journal of Solid State Circuits, 33, No. 5, pp. 676-686, May 1998.

[5] A. Wassatsch, D. Timmermann, "DYNAMIC- A Java Based Toolset For Integrating Dynamic Logic Circuits Into A Standard VLSI Design Flow", International Cadence User Group Conference (ICU'2000), San Jose (CA) USA, pp. SIG IC - ic6, September 2000.

[6] A. Wassatsch, D. Timmermann, "Untersuchung zum Einflub der Speziellen Anforderungen dynamischer Schaltungstechnik auf den Systementwurf", ITG/GI/GMM Workshop: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, Frankfurt/Main(Germany), VDE-Verlag, S. 278-287, 28.2.-1.3. 2000.