Design and ASIC Implementation of Triple Data Encryption and Decryption Standard Algorithm

Vijayabhaskararao.Manda[1] Bhavana P.Srivastava[2]

Abstract

The main objective of this paper is to evaluate the performance of the algorithm TDES (Triple Data Encryption Standard), used for encryption and decryption on high speed secure data transmission. The algorithm is designed and implemented for ASIC model through a hardware description language, referred as VHDL (Very High Speed Integrated Circuit Hardware Description Language). Security issues are playing dominant role in today's high speed communication systems. Every single transmitted bit of information needs to be processed into an unrecognizable form in order to be secured. This enciphering of the data is necessary to take place in real time and for this procedure cryptography is the main mechanism to secure digital information.

Triple DES is based on the DES algorithm. It is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the attacks that can be used to reduce the amount of time it takes to break DES. However, even this more powerful version of DES may not be strong enough to protect . for very much longer. The procedure for encryption is exactly the same as DES, but it is repeated three times. Hence it is named as Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key. Cryptography Algorithm DES contains two processes like encryption and decryption. Encryption process and decryption process both works on same algorithm but they vary in the application of key. Key is placed from 1 to 16 in encryption and 16 to 1 in decryption. These two processes can be executed efficiently with the help multiplex based design. Which is incorporated at key forcing .This design is capable of reducing the hardware area to 50% when compared to the conventional approach.

Key Words: DES, Triple DES, Cryptography, Encryption, Decryption, Block cipher

1. Introduction

Triple Data Encryption Standard (DES) processor is fast data encryption is becoming a more important requirement for applications such as secure networking. Using DES cannot always ensure high security; therefore the data can be encrypted three times with the same algorithm to realize Triple DES. Through the utilization of dynamic TSMC, 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 syntheses and the verification. We conclude with a discussion of the results and estimate comparisons with other chips.

[1&2] Dept. of Electronics and Communication Engineering Maulana Azad National Institute of Technology, Bhopal-M.P Email: friends_vijay143@yahoo.co.in, sonibhavana1@gmail.com

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-bit 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 initial permutation (IP). The key-dependent computation can be defined in terms 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 bit input block is divided into two halves, each consisting of 32 bits. One half is used as input to the function F, and the result is exclusive 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 bit binary number, represented as 0-15. A particular entry in a single S-box is selected in six bits; 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 bit input to a 48 bit output that is added mod two to the round key. The permutations in the key-schedule, PC1 and PC2, inter mix 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 × 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 Figure 1, forward and inverse cipher operation is a compound operation of DEA forward and inverse transformations as specified.

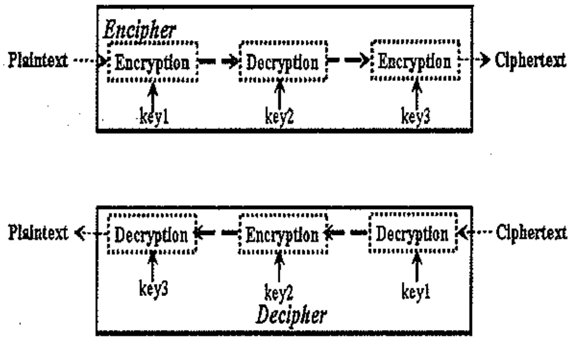


Figure 1 TDEA

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.

Let FKeyX (d) and IKeyY (d), respectively, represent the DEA forward and inverse transformations on data d using key bundle KEY. The following operations are used:

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

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

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

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

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

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

2. 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.

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:

1. Be secret

2. Be generated randomly or pseudo randomly

3. Be independent of other key bundles

4. 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;

5. Be used in the appropriate order as specified by the particular mode

6. 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.

2. Difference between DES, AES and Triple DES

	DES	AES	TDES
Key	56	128	168
Cycle	16	10	48
Frequency	288MHz	172MHz	311.5MHz
Area	Small	More	Small
Efficiency	16.5	60.3	49.4
S-box	Table	Table	Table
Technology	180nm	180nm	180nm

Table 1 : Difference of Cryptography

The Table 1 shows difference between DES, AES and TDES algorithm. Based on Table 1 TDES algorithm gives the less area and less power and less same efficient of AES algorithm. The Key length is more than DES and AES algorithm. So this will gives the more secure than DES algorithm. The pipelining is more than DES and TDES algorithm.

3. Architecture Overview of Encryption Algorithm

DES Figure 2 shows the single DES algorithm is a block cipher: It encrypts/decrypts data in 64-bit blocks using a 64-bit key (although it's effective key length is in reality only 56-bit). DES is a symmetric algorithm: The same algorithm and key are used for both encryption and decryption. DES is an iterative cipher: the basic building block (a substitution followed by a permutation) called a round is repeated 16 times. For each DES round, a sub-key is derived from the original key using an algorithm called key schedule.

Key schedule for encryption and decryption is the same except for the minor difference in the order (reverse) of the sub-keys for decryption. A basic algorithm flow for encrypting/decrypting one block of data is shown in Figure 2. Encryption begins with an initial permutation (IP), which scrambles the 64-bit plain-text in a fixed pattern. The result of the initial permutation is sent to two 32-bit registers, called the right half register and left half register. Those registers hold the two halves of the intermediate results through successive 16 iterations.

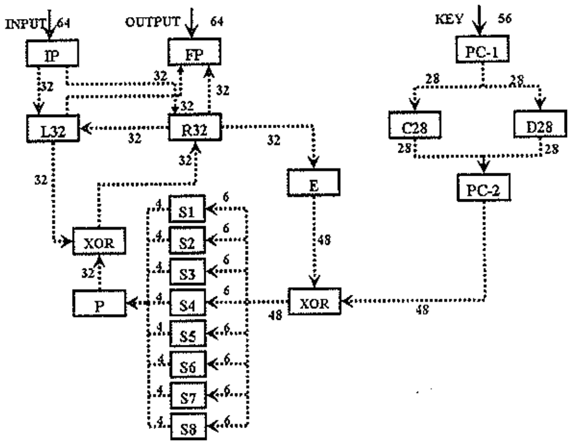


Figure 2 DES Algorithm

The contents of the right half register are permuted (permutation E) and sent to an exclusive-OR unit along with the sub-key for each iteration. Note that some bits are selected twice, allowing the 32-bit register to expand to 48 bits. The 48-bit output of the exclusive-OR block is divided into eight groups (6-bits each) to address eight substitution memories (S-boxes).

A permutation P is applied to 32-bit output from S-boxes and then feed into an exclusive-OR block along with the contents of the left half register. The output of this block is written into a temporary register, concluding the first iteration. At the next clock cycle, the contents of the temporary registers are written into the right half register and previous contents of the right half register are written into left half register.

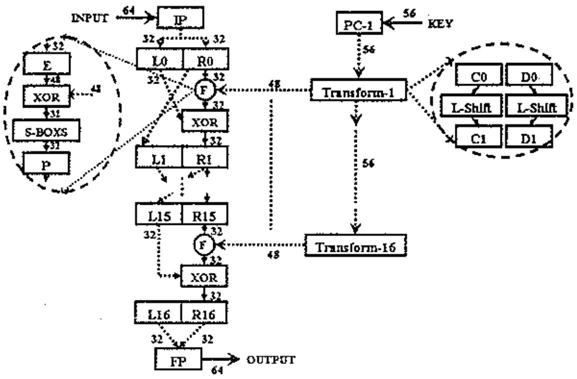


Figure 3 Encryption Pipelining

This process is repeated through the whole 16 DES iterations figure 3 (pipelining process). After 16 iterations, the right half and left half register contents are subjected to a final permutation IP"1, which is the inverse of the initial permutation. The output of IP"1 is the 64-bit cipher-text.

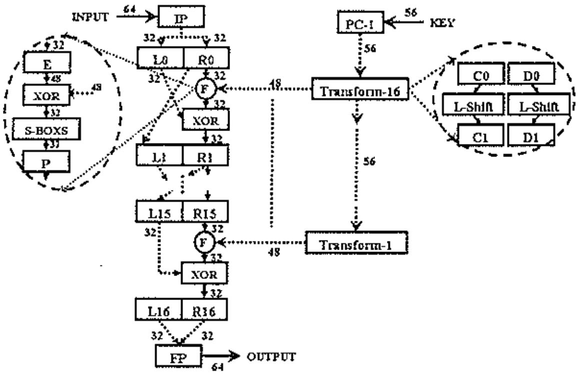


Figure 4 Decryption Pipelining

Figure 4 shows the single decryption algorithm. This is reverse process for the single encryption algorithm. This algorithm shows varies level for the design. The difference between the encryption and decryption process is the key scheduling is different. The encryption algorithm the pipeline process is from 1 to 16. But the decryption process is from 16 to1.

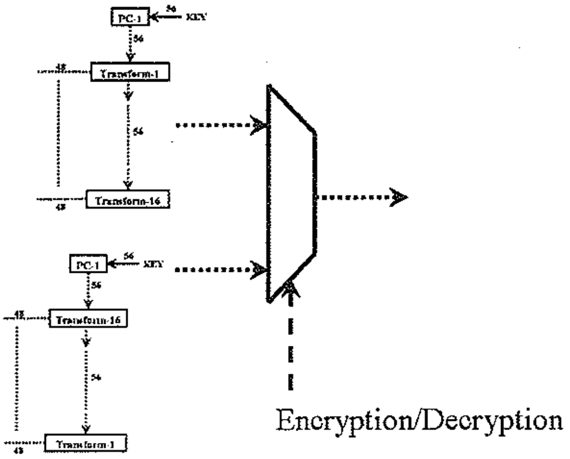


Figure 5 Mux Based Design

The Figure 5 shows the multiplexers based design. This type of design is reducing the area and power of the design. Based on constrains the frequency of the operation the design will increased.

IV Proposed TDES Algorithm

The Figure 6 shows the proposed Triple DES Algorithm. This algorithm is reducing the area as will as power also. Because of the design is reduce the almost 50%. So it should reduce the power and cell in the over all design. The design will finding the functionality will base on the sequence of the signals and input bits. Figure 4.6 gives varies keys used decryption the input data. Clock is positive edge, Reset is active low, Input enable is active high, Serial input data bits is 0123456789abcdef, Multiplexer based selecting keys Key1:133457799bbcdf (00), Key2:3457799bbcdff1 (01), Key3:0abcdabcdabcdc (01), Output enable -active high, Serial output:Encryption:23feab48179a2b4c, Decryption:0123456789abcdef.

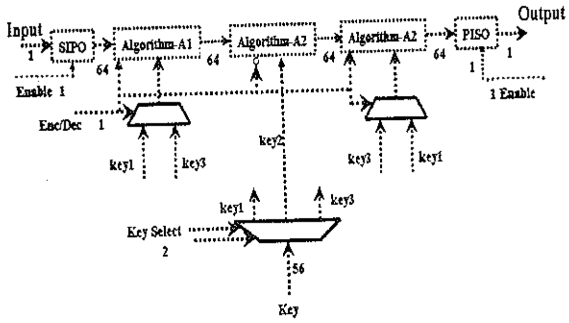


Figure 6 Proposed TDES Algorithm

V. Results

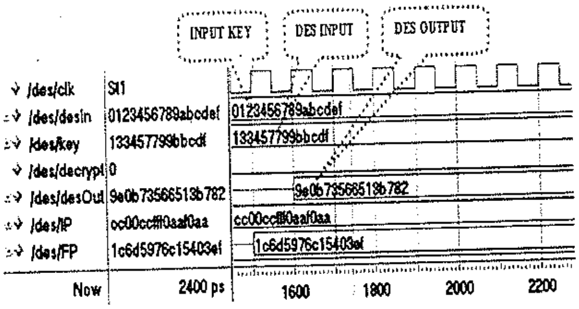

INPUT KEY
DES INPUT
DES OUTPUT
/des/clk St1
/des/desIn 0123456789abcdef 0123456789abcdef
/des/key 133457799bbcdf 133457799bbcdf
/des/decrypt 0
/des/desOut 9e0b73566513b782 9e0b73566513b782
/des/IP cc00ccfff0aaf0aa cc00ccfff0aaf0aa
/des/FP 1c6d5976c15403ef 1c6d5976c15403ef
Now 2400 ps 1600 1800 2000 2200

Figure 7 Simulation Results for Single Encryption

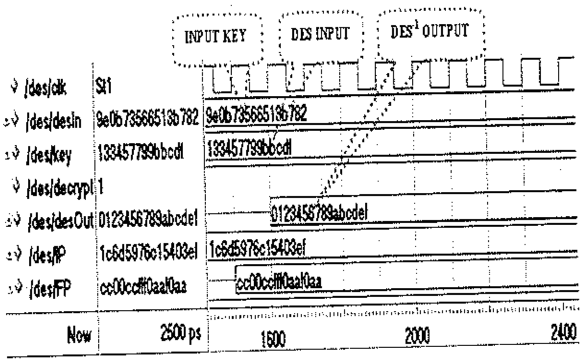

INPUT KEY
DES INPUT
DES^1 OUTPUT
/des/clk St1
/des/desIn 9e0b73566513b782 9e0b73566513b782
/des/key 133457799bbcdf 133457799bbcdf
/des/decrypt 1
/des/desOut 0123456789abcdef 0123456789abcdef
/des/IP 1c6d5976c15403ef 1c6d5976c15403ef
/des/FP cc00ccfff0aaf0aa cc00ccfff0aaf0aa
Now 2500 ps 1600 2000 2400

Figure 8 Simulation Results for Single Decryption

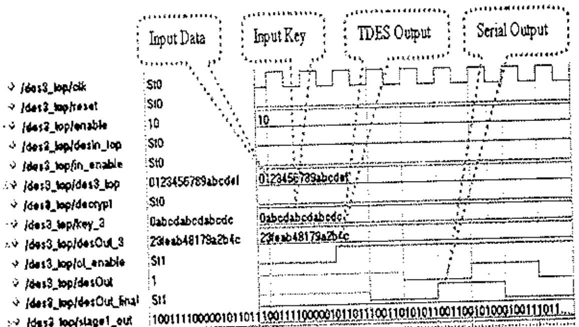

Input Data
Input Key
TDES Output
Serial Output
/des3_top/clk St0
/des3_top/reset St0
/des3_top/enable 10 10
/des3_top/desIn_top St0
/des3_top/in_enable St0
/des3_top/des3_top 0123456789abcdef 0123456789abcdef
/des3_top/decrypt St0
/des3_top/key_3 0abcdabcdabcdc 0abcdabcdabcdc
/des3_top/desOut_3 23feab48179a2b4c 23feab48179a2b4c
/des3_top/ol_enable St1
/des3_top/desOut 1
/des3_top/desOut_final St1
/des3_top/stage1_out 1001111000001011011 10011110000010110111001101010110011001010001001111011...
Now 23500 ps 22800 23200

Figure 9 Simulation Results for Triple Encryption

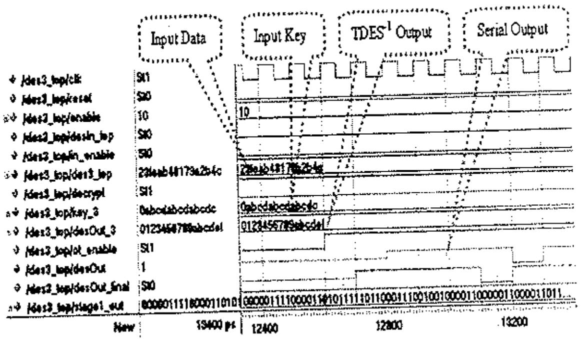

Input Data
Input Key
TDES^1 Output
Serial Output
/des3_top/clk St1
/des3_top/reset St0
/des3_top/enable 10 10
/des3_top/desIn_top St0
/des3_top/in_enable St0
/des3_top/des3_top 23feab48179a2b4c 23feab48179a2b4c
/des3_top/decrypt St1
/des3_top/key_3 0abcdabcdabcdc 0abcdabcdabcdc
/des3_top/desOut_3 0123456789abcdef 0123456789abcdef
/des3_top/ol_enable St1
/des3_top/desOut 1
/des3_top/desOut_final St0
/des3_top/stage1_out 0000011110000110101 0000011110000110101111011000111001001000011000001100001011...
Now 13400 ps 12400 12800 13200

Figure 10 Simulation Results for Triple Decryption

The Figures 7,8,9 and10 shows the single encryption, single decryption, triple encryption, and triple decryption simulation result. The results area verified using different test cases in the design. For simulation ModelSim tool are used.

The RTL code is synthesized using Design Compiler from Synopsys. The tool reads the HDL code and translates the logic to logic equations which are mapped to the cells in the linked libraries, with logic level optimization. Further on the basis of constraints the tool proceeds to provide the optimized gate level netlist. Then the compiled results are checked for the setup and hold violations i.e. the slack is met or not. If not met the constraints are again changed as per requirement. The design timing verification is checked again. The timing report of design compiler is as below. The libraries used are tcb013ghplvtwc.db, tcb013ghplvtbc.db, tcb013ghphvtwc.db and tcb013ghphvtbc.db. These libraries are set as link and target library.

This provides the timing for the critical path for setup timing by default and for hold by using the switch –delay min. The time race for data and clock for the arrival time (AT) and required time (RT) are shown with all device delays involved including the external delays, clock uncertainties, setup /hold timings. Slack is checked to be positive with margin for DFT and also some margin for static timing analysis. The AT for setup timing analysis is 5.04 with slack 0.07. The RT for hold timing analysis is 0.49 with slack 0.00.

The report provides the complete area 553311.66 units and its break-up to combinational (389726.12), non-combinational (163583.5533123) and interconnect (the wire load has zero net area). It also provides the list of libraries used in the design and number of ports, nets, cells and references of the design

This in addition to the libraries used displays the operating conditions, wire load mode for the library and wire load model for each reference. It provides the cell internal power (the device power – 58.3984mW), the switching power (17.4733mW) with total dynamic power 17.4733mW and the leakage power (1.2033mW).

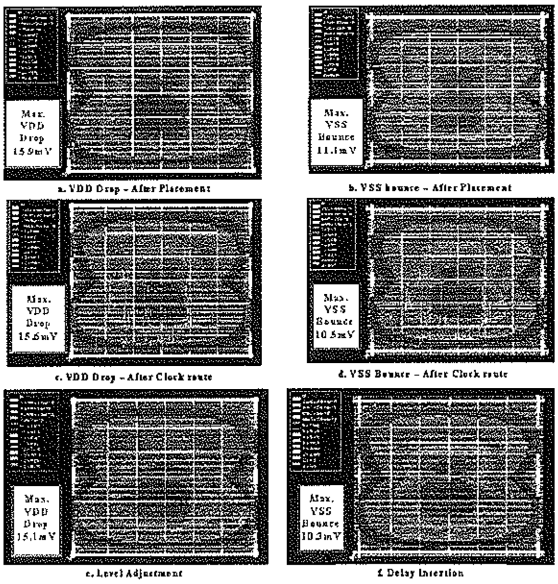
	Full design	Optimized design
Cells	113688	57385
Area	1338798mm^{2}	553311.66mm^{2}
Timing	5.5ns	5.5ns
Scan Paths	14	8
Flip-flops	12279	6327
Faults Coverage	803191	404938
Test Coverage	100%	100%
Switching Power	2.363mw	1.196mw
Technology	130nm	130nm

Table 2 Difference of Full and Optimized Design

The table 2 shows the difference between the full design and optimized design in an ASIC front-end flow. This net-list is taken in to the ASIC back-end flow.

Floor planning could be done on the basis of Aspect ratio or specifying height and width directly. During the floor planning the Core Utilization, Aspect Ratio and Row/ Core utilization parameters are defined. Total number of cell instances is 57385.Total number of nets is 57469.Total number of ports is 81. The Core utilization is 0.683.Number Of Rows is 27. Core Width is 999.58.Core Height is 999.99.Aspect Ratio is 1. from Synopsys routed design using Astro tool.

The total Io net switching power is 1.62048 mw. Total switching power is 21.838 mw. Total short-circuit power is 37.9524 mw. Total internal power is 15.588 mw. Total leakage power is 1.16376 mw. Total power is76.5422 mw.



a. VDD Drop – After Placement

b. VSS bounce – After Placement

c. VDD Drop – After Clock route

d. VSS Bounce – After Clock route

e. Level Adjustment

f. Delay Insertion

Figure 11 IR Drop Analysis

The figure 11 show's varies stage IR drop analysis in this design using Astro Rail tool from synopsys.

VI. Conclusion

The DES algorithm is implemented in HDL level. The encryption and decryption algorithm an implemented in HDL coding. DES and TDES architecture are designed using key scheduling algorithm to reduce the area and power in the design. DES and TDES are verified for its functionality with the following test vectors (Input data, 3keys, enable signals). The design is synthesized using 130nm technology and has the following specification (Pins 81, total cell 57385, total nets 57469, and total register 6327). The optimized design occerfied the less area (553311mm2), high frequency (181.1 MHz), less power (76.5422mw), less IR drop (1.4%) and 100% test coverage in 130nm technology. RTL to GDSII design and verified at various levels for the proposed design is completed with the following specification (Total Input pins 74, total Output pins 8).

References

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

[2] Arich.T, and Eleuldj.M, *"Hardware implementations of the data encryption standard"*, 14th International Conference on Microelectronics (ICM), 2002, pages 100- 103, December 2002

[3] Mostafa-Sami M. Mostafa, Safia.H.Deif 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, Volume 26 Issue 28, Pages 137 – 142, April 2008

[4] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, *"Highperformance microprocessor design"*, Journal of Solid State Circuits, Vol. 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 Einfluß 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.

[7] I. Kim, C. S. Steele, J. G. Koller, *"A Fully Pipelined, 700MByte/s DES Encryption Core"*, Proceedings of the 9th IEEE Great Lakes Symposium on VLSI, March 4-6, 1999.

[8] J. Yuan, I. Karlsson and C. Svensson, *"A True Single Phase Clock Dynamic CMOS Circuit Technique"*, IEEE Journal of Solid-State Circuits, Vol. SC-22, 1987, pp. 899- 901.

[9] Synopsys, Inc., 700 East Middlefield Road, CA 94043-4022 United States of America. Synopsys Synthesis and Simulation Tools, 2000 edition.

[10] Austria Micro Systeme International AG (AMS), Austria, 0.6 um CMOS process cua, v3.12.

[11] National Bureau of Standards FIPS Publication 46, *"DES modes of operation"*, 1977.