An efficient design for Data Encryption and Decryption using Reconfigurable Reversible Logic Gates

Patancheru Jyothi[1], B. Kalpana[2]

[1]Student, Master of Technology, Dept. of Electronics and communication Engineering, Mallareddy College of Engineering (Autonomous), Telangana, India.
[2]Assistant Professor, Dept. of Electronics and communication Engineering, Mallareddy College of Engineering (Autonomous), Telangana, India

---***---

Abstract - *The development in the field of nanometer technology leads to minimize the power consumption of logic circuits. The reversible logic design has been one of the promising technologies gaining greater interest due to less dissipation of heat and low power consumption. On the other hand, energy dissipation in reversible logic gates can decrease to zero. Recently an approach to encryption based on using reversible logic circuits is proposed. This paper presents a solution for designing data encryption and decryption schemes based entirely on reconfigurable reversible logic. In our solution, a building block of encryption and decryption scheme is a cascade of 4-input reversible gates. In this way, the building block can perform any reversible 4-variable function. For this purpose, a reconfigurable reversible logic gate has been proposed. The design of such a reconfigurable reversible gate is built from standard reversible gates, i.e., NOT, CNOT, Fredkin and Toffoli gates. This paper presents a complete scheme for data encryption and decryption of 9-bit data using Verilog HDL language. Simulation and verification of this scheme are obtained on Xilinx ISE 14.7.*

Key Words**: Encryption, Decryption, Reversible circuits, Reconfigurable reversible gate, logic gates.**

1. INTRODUCTION

The developing technologies have increased the demand for high-performance computing. According to G. Moore's law, some transistor counts to be integrated per unit area in devices will almost double in one and half year. To accomplish fast computation, high packaging density in the logic circuits is required which brings about more heat dissipation. The conventional computing is found unable to deal with low power, high compaction and heat dissipation issues of the current computing environment. Recently, it is applied to cryptography. A reversible gate is a one-to-one correspondence between its inputs and outputs. Research on reversible logic circuits is motivated by advances in quantum computing, nanotechnology, and low-power design. As a result, reversible logic synthesis has been intensively studied recently.

The attention is focused on the synthesis of circuits built from the NCT library of gates, i.e., NOT, CNOT and Toffoli gates. Recent simulation tools based on FPGAs have enabled modeling of such circuits. In the paper, we study an application of reversible logic to developing encryption and decryption circuits. The simple implementation of a cipher using reversible logic circuits was the aim of this work. Each gate used in a cascade of reversible gates is determined by the main key. By choosing different main keys, different cascades and different substitution, data encryption and decryption are determined. For this purpose, a reconfigurable reversible logic gate has been proposed. Results of Xilinx ISE based simulation of a simple data encryption and decryption circuits built from reconfigurable reversible logic gates are also presented in the paper.

1.1. Reversible Definitions

Let us recall basic notions of reversible logic gates and circuits.

Definition-1: A completely determined n-input and n-output Boolean function (referred to as n*n function) are reversible if it maps each input value to a unique output value.
For n*n Boolean functions, there are 2n! Reversible logics.
For n = 3 this number equals to 40,320, for n = 4 is greater than 2Â1013.

Definition-2: An n-input, n-output (n*n) gate (or circuit) is reversible if it realizes an n*n reversible function. In a reversible logic circuit fan out always equal to 1. As a result, n*n reversible circuits can be built only as a cascade of k*k reversible gates (k ≤ n).

Definition-3: A set of reversible gates which can be used to build reversible circuits is called a gate library.
The so-called NCT library for n ≤four consists of 1*1 NOT, 2*2 CNOT and 3*3 and 4*4 TOFFOLI gates.
Definitions of mixed polarity generalizations of these gates as well as 3*3 Fredkin gate are given below.

Definition-4: Let ai ∈ {0, 1} for i = 1, 2, 3, 4, and let x' denotes negation of x.
1*1 NOT gate (x1) performs the operation
(x1) - (x1 ⊕ 1),
2*2 CNOT (x1, x2) gate performs the operation

$(x1, x2) - (x1, (x1 \oplus a1) \oplus x2)$,
3*3 TOFFOLI(x1, x2, x3) gate performs the operation
$(x1, x2, x3) - (x1, x2, (x1 \oplus a1)$
$(x2 \oplus a2) \oplus x3)$,
4*4 TOFFOLI4(x1, x2, x3, x4) gate performs the operation
$(x1, x2, x3, x4) - (x1, x2, x3, (x1 \oplus a1)(x2 \oplus a2)$
$(x3 \oplus a3) \oplus x4)$,
3*3 FREDKIN(x1, x2, x3) gate performs the operation
$(x1, x2, x3) - (x1, x1' x2 + x1 x3, x1 x2 + x1' x3)$.

The above defined i*i gates, where i = 1, 2, 3, 4, (also, denoted by N, C, T, T4 and F, respectively) invert input xi, if and only if the values of inputs x1, x2, ..., xi-1 differ from corresponding a1, a2, ..., ai-1 coefficients, passes these inputs that are unchanged to corresponding outputs. Control lines are the signals which are passed unchanged from input to output of the gate.

Target is the signal 'xi' which can be modified by the gate. In the Fredkin gate, x1 is the control line. When its input signal is 0 then the signals x2 and x3 are passed to outputs unchanged, and when it's input signal is one then the signals x2 and x3 are swapped.

If all a1, a2..., ai coefficients equals zero, that gate has positive-polarity control lines. On the other hand, if all a1, a2..., ai coefficients equals one, the gate has negative polarity control lines. These names are obtained after expanding the expressions for the functions realized by the gate.

Positive polarity means that all inputs which correspond to control lines directly affect the target line. Negative-polarity means that the target line is affected only if the values of control lines are equal to 0. The term mixed-polarity control lines are used, if all values of the coefficients a1, a2, ..., ai allowed to be either 0 or 1. Pictorial symbols of the above defined positive-polarity gates are shown in Figure 1, and the pictorial symbol of Fredkin gate is shown in Figure. 2.

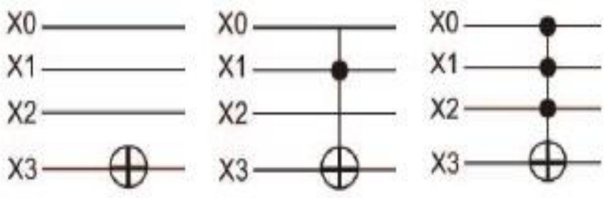


Figure 1: The Graphical representations of reversible gates N3, C3-1 and T3-210.

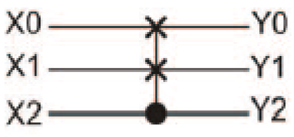


Figure 2: The Graphical representation of Fredkin gate F10-2.

In the description of a gate from NCT library, e.g., T3-210 (see Figure 1), the first number appearing after the gates name (in this case: T) denotes the target line (3) while the numbers are appearing after the hyphen denote control lines (2, 1, 0).

In the description of a Fredkin gate, e.g, F10-2, the first two numbers appearing after the gates describes name (in this example: F) denote controlled (swapping) lines (1 and 0) while the number appearing after the hyphen denotes control line (2).

The number of positive-polarity gates belonging to NCT library which can appear in 4-line circuits is 32 (4 NOT gates, 12 CNOT gates, 4 T4 gates and 12 Toffoli gates).

2. RRG IMPLEMENTATION

Reconfigurable reversible gate replaces one of 32 gates from the NCT library in the cipher. This is why, besides four inputs transferring information data, there have to be five lines for choosing one of 32 types of gates.

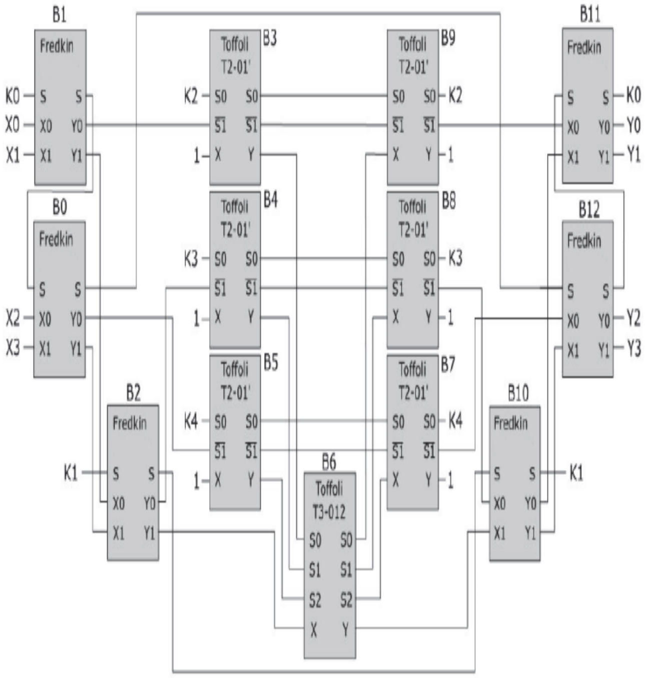


Figure 3: The General scheme of Reconfigurable Reversible Gate (RRG).

At the present state-of-the-art of reversible logic synthesis, it is not possible to optimally synthesize 9-input reversible gate. After considering many variants of gates RRG the circuit is shown in Figure 3. It has been constructed with the relatively small quantum cost equal to 79.

The RRG built from basic reversible gates is shown in Figure 4.

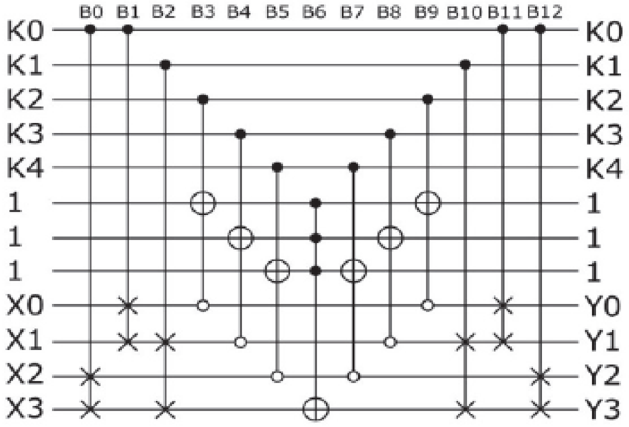


Figure 4: Reconfigurable Reversible Gate (RRG).

The first three Fredkin gates B0, B1 and B2 select the input signal which will be modified by the 4- Input Toffoli gate B6. The next three 3-input Toffoli gates B3, B4 and B5 feed control signals for the gate B6 that are selected, by the given configuration either input or constant signals.
Thus, the functioning of the gate B6 is determined by signals K. Gates B7, B8 and B9 reconstruct constant signals, while gates B10, B11 and B12 fix the order of output signals Y3, Y2, Y1, and Y0.

For simulation of a reversible gate, a procedure B_FREDKIN and B_NCT date from NCT library have been constructed in VHDL and shown in Tables I and II.

2.1 PROCEDURE B_FREDKIN

In the procedure B_NCT signals C1, C2 and C3 determine the number of lines to which control inputs of Toffoli gate are attached while the number of controlled lines is determined by signal C0.

```
Procedure B_FREDKIN

(X:       in std_logic_vector; -- input line

C: in integer; -- number of control line

L1: in integer; -- number of first line

L2: in integer; -- number of second line

Y:        out std_logic_vector) -- output line
```

Binary signals N1, N2, and N3, determine the polarization of the control signal Ci. Value N1 = 1 indicates that the signal is a negative control signal.

2.2 PROCEDURE B_NCT

The above-presented procedures were used for describing the gate RRG as shown below.

```
KX(0):=(KEY(0)&KEY(1)&KEY(2)&KEY(3)&KEY(4)&"111"&
DIN(0)&DIN(1)&DIN(2)& DIN(3));
B_FREDKIN (KX (0), 0, 10, 11, KX (1));
B_FREDKIN (KX (1), 0, 8, 9, KX(2));
B_FREDKIN (KX (2), 1, 9, 11, KX (3));
B_NCT (KX (3), 2,'0', 8, '1', 12,'0', 5, KX (4));
B_NCT (KX (4), 3, '0', 9, '1', 12,'0', 6, KX (5));
B_NCT (KX (5), 4, '0', 10, '1', 12,'0', 7, KX (6));
B_NCT (KX (6), 5, '0', 6, '0', 7,'0', 11, KX (7));
B_NCT (KX (7), 4, '0', 10,'1', 12,'0', 7, KX (8));
B_NCT (KX (8), 3, '0', 9,'1', 12,'0', 6, KX (9));
B_NCT (KX (9), 2,'0', 8 ,'1' , 12 , '0' , 5 ,KX(10));
B_FREDKIN (KX (10), 1, 9, 11, KX (11));
B_FREDKIN (KX (11), 0, 8, 9, KX (12));
B_FREDKIN (KX (12), 0, 10, 11, KX (13));
```

12-bit word KX is the state of line just before or just after gate and is transferred from i-th to (i+1) th gate.

```
Procedure B_NCT

(X:       in std_logic_vector ; -- input line

C1: in integer; -- control line

N1: in std_logic; -- negative control line

C2: in integer; -- control line

N2: in std_logic; -- negative control line C2

C3: in integer; -- control line

N3: in std_logic; -- negative control line C3

C0: in integer; -- control line output

Y: out std_logic_vector) – output line

begin

If C1=12 then Y: =X; Y (C0): not X (C0);
```

3. DESIGN OF ENCRYPTION BY RRG GATE

The general idea of a cipher built from reversible gates was presented Figure 5.

The basic element of the cipher is a cascade of 16 4-input reversible gates. The same main key is used for data encryption and decryption. The order of gates in the cascade for decryption is reversed in comparison with the cascade for encryption that ensures that it transforms cipher-text into plaintext.

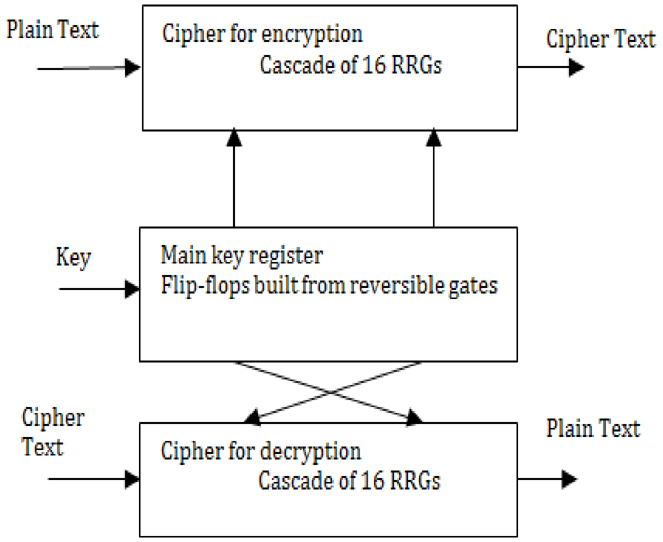


Figure 5: The General idea of a cipher for encryption and decryption

It has been proved that optimal circuits for any of the 16! (Equal to over 2x1013) require at most 15 4-input gates.

Thus the circuit shown in Figure 6 enables the realization of any 4-variable reversible function. The circuit has K which denotes 80 inputs that are partitioned into groups with five inputs in each of them.

A 5-line group K [(5*(i+1) – 1):5*i] is used to configure with RRG gate. All inputs K are transferred to outputs so they can be reused for controlling the next gate.

There are 4 data inputs X [3:0] and three lines with constant inputs (equal to 1) and equivalent outputs.

A detailed description of the main key register, as well as the circuit modifying its contents, during encryption and decryption, are presented in following figures.

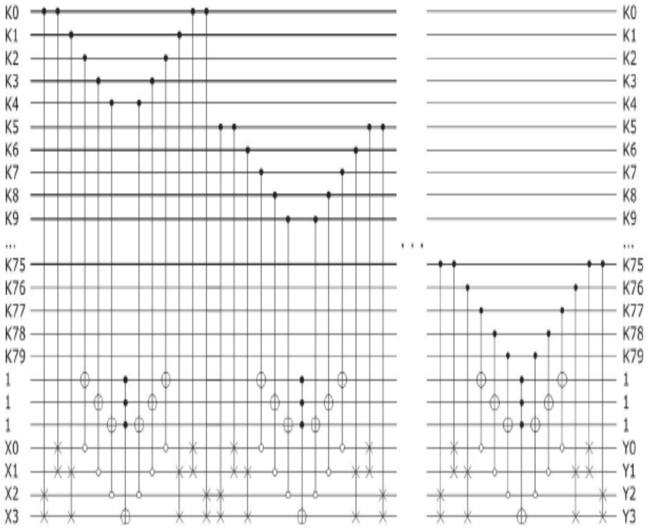


Figure 6: The cascade of reversible gates implementing the 4-bit cipher.

4. SYNTHESIS AND SIMULATION RESULTS

For encryption and decryption, a 5-bit key, two 4-bit ciphers and the main key, a register was used. In this section, first, we will see the synthesis and simulation of the Encryption and Decryption using a reconfigurable reversible gate. Encryption and Decryption using reconfigurable reversible gates are designed on Xilinx ISE 14.7 with Verilog HDL. The RTL schematics and simulation results of the proposed design are shown below.

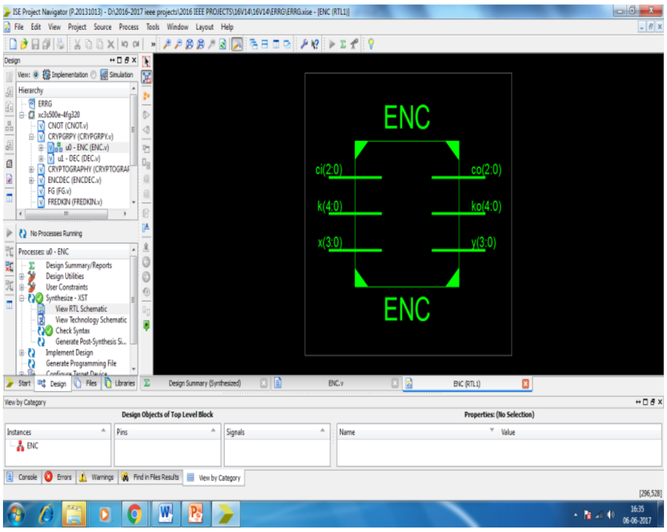


Figure 7: RTL Schematic of Proposed Reconfigurable Reversible Encryption

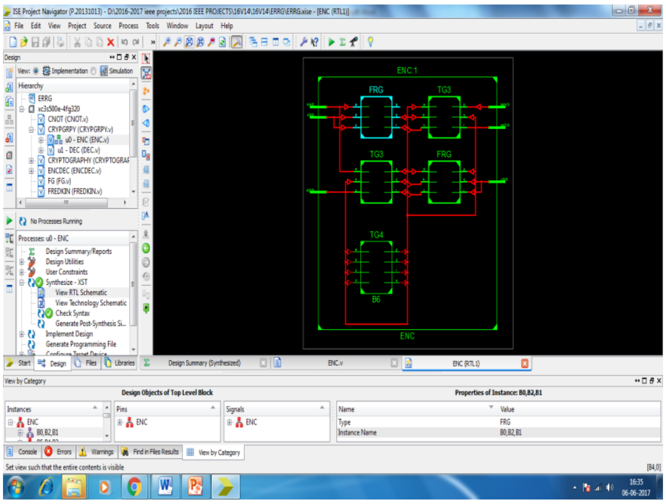


Figure 8: Internal RTL Schematic of Proposed Reconfigurable Reversible Encryption

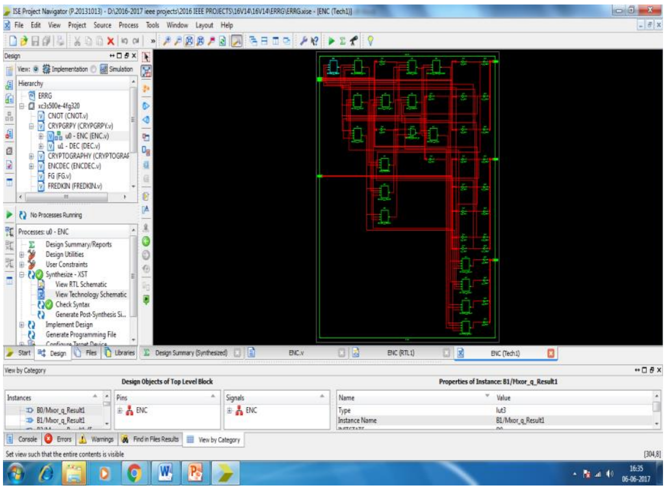
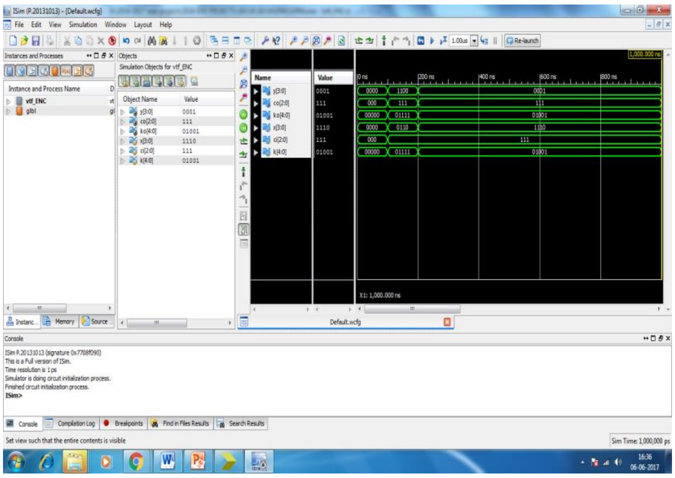


Figure 9: Technology Schematic of Proposed reconfigurable reversible Encryption and Decryption



Simulation Objects for vtf_ENC

Instance and Process Name
vtf_ENC
glbl

Object Name	Value
y[3:0]	0001
co[2:0]	111
ko[4:0]	01001
x[3:0]	1110
ci[2:0]	111
k[4:0]	01001

```
ISim P.20131013 (signature 0x7708f090)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Set view such that the entire contents is visible
Sim Time: 1,000,000 ps

Figure 10: Simulation of Proposed Reconfigurable Reversible Encryption

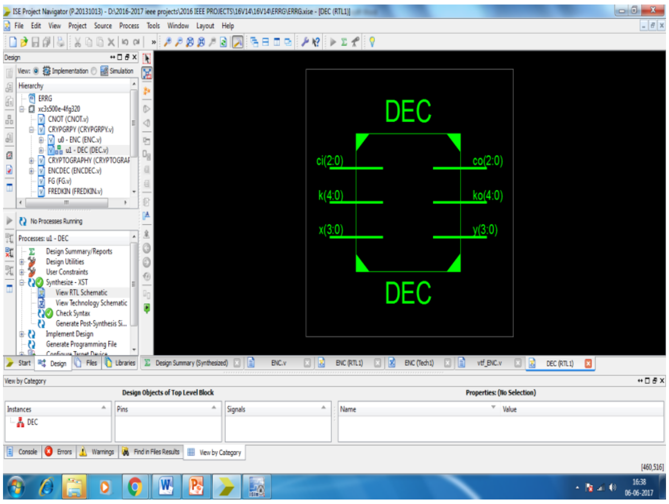


Figure 11: RTL Schematic of Proposed Reconfigurable Reversible Decryption

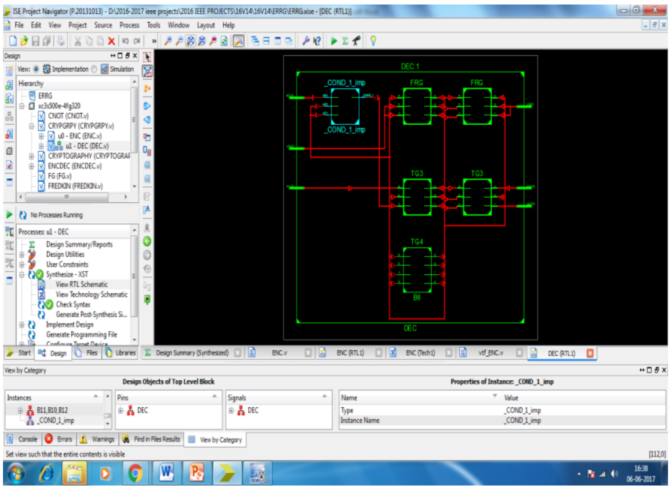


Figure 12: Internal RTL Schematic of Proposed Reversible Decryption

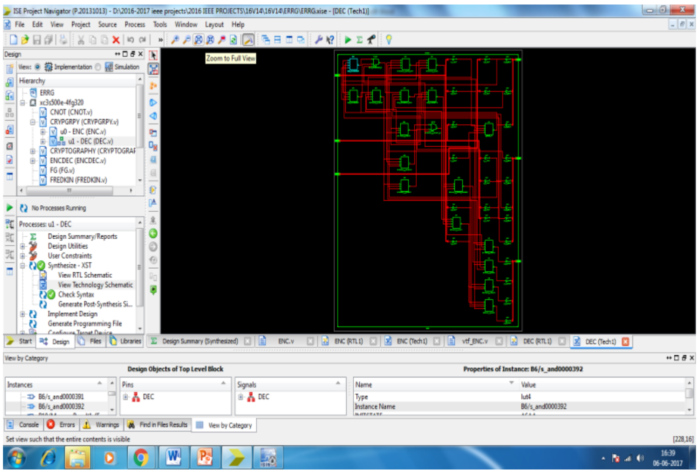
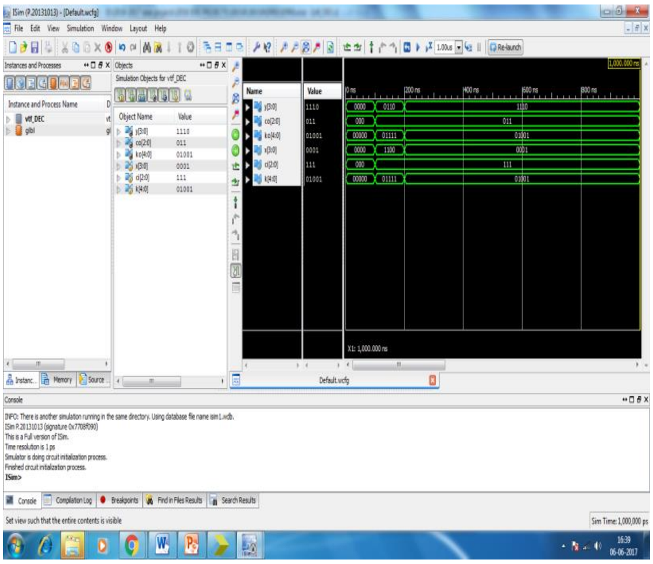


Figure 13: Technology Schematic of Proposed Reconfigurable Reversible Decryption



Object Name	Value
y[3:0]	1110
co[2:0]	011
ko[4:0]	01001
x[3:0]	0001
ci[2:0]	111
k[4:0]	01001

```
INFO: There is another simulation running in the same directory. Using database file name isim1.wdb.
ISim P.20131013 (signature 0x7708f090)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Set view such that the entire contents is visible

Sim Time: 1,000,000 ps

Figure 14: Simulation of Proposed Reconfigurable Reversible Decryption

5. CONCLUSION

Firstly, we have coded Verilog code for the reconfigurable reversible data encryption and decryption, and all the synthesis and simulation results are implemented on Xilinx ISE 14.7. The main aim of this paper is a design of simple reconfigurable reversible gate (RRG) which enables implementation of any of the 32 4-input reversible gates from the NCT library. An application of this gate is to implement ciphers for encryption and decryption in the form of binary data. Results of data encryption and decryption simulation of the cipher built from reversible gates are also presented.

REFERENCES

[1] A. De Vos, Reversible Computing. Fundamentals, Quantum Computing, and Applications, 2010.

[2] H. Thapliyal and M. Zwolinski,: "Reversible logic to cryptographic hardware" Proc. 49th International Midwest Conference on Circuits and Systems,2006.

[3] N. M. Nayeem, L. Jamal, and H. M. H. Babu, "Efficiently reversible multiplier and its application to hardware cryptography," Journal of Computer Science, 2009.

[4] Y. Zhang, Z. Guan, and Z. Nie, "Function modular design of DES encryption system based on reversible logic gates," Proc. International Conference on Multimedia Communications, 2010.

[5] A. Banerjee, "Reversible cryptographic hardware with optimized quantum cost and delay," Proc. Annual IEEE India Conference, pp. 1- 4, 2010.