

Encryption Using Reconfigurable Reversible Logic Gate and Its Simulation in FPGAs

Marcin Bryk¹, Krzysztof Gracki¹, Paweł Kerntopf², Marek Pawłowski¹, Andrzej Skorupski¹

¹Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland

²Faculty of Physics and Applied Informatics, University of Łódź, Łódź, Poland

m.pawlowski@ii.pw.edu.pl, p.kerntopf@ii.pw.edu.pl

Abstract—Recently an approach to encryption/decryption based on using reversible logic circuits has been proposed. The reason for this is that conventional microelectronic technologies are reaching their limits. On the other hand, reversible logic circuits can decrease energy dissipation theoretically to zero. This paper presents a solution to designing encryption schemes based entirely on reversible logic. In our solution a building block of an encryption 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 gate has been proposed. The design of such a reconfigurable gate built from standard reversible gates, i.e. NOT, CNOT, Toffoli and Fredkin gates, is presented. In the paper a complete scheme for encryption/decryption of 8-bit data is described using VHDL language and its quantum cost is calculated. Simulation and verification of this scheme in FPGAs conclude the paper.

Index Terms—encryption, reversible logic circuits, reconfigurable reversible gate, FPGA

I. INTRODUCTION

Reversible computation is an emerging research area. It has applications in many areas of computer science, e.g. quantum computing, nanotechnologies, optical computing, digital signal processing, bioinformatics and in low power computation [1]. Recently, it has been applied to cryptography [2-10].

A circuit (gate) is called reversible if there 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. Therefore, reversible logic synthesis has been intensively studied recently. The attention is focused mainly on the synthesis of circuits built from the NCT library of gates, i.e. NOT, CNOT and Toffoli gates. Modern simulation tools based on FPGAs have enabled modeling of such circuits [11]. The design of a cipher in VHDL was described [9].

In the paper we study an application of reversible logic to developing encryption circuits. The simple implementation of a cipher using reversible 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 encryption are determined. For this purpose a reconfigurable reversible gate has been proposed. The design of this gate is presented for the first time in the literature. Results of FPGA-based simulation of a simple encryption circuit built from reversible gates are also presented.

The paper is organized as follows. Section 2 recalls basic concepts of reversible logic. In Section 3 design of reconfigurable reversible gate (RRG) is introduced. In Section 4 an implementation of encryption/decryption by cascades of RRG gates is given. In Section 5 simulation of the cipher in FPGAs is presented. Section 6 summarizes the paper.

II. PRELIMINARIES

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

Definition 1. A completely specified n -input n -output Boolean function (referred to as n^*n function) is called *reversible* if it maps each input assignment into a unique output assignment.

There are $2^n!$ reversible n^*n Boolean functions. For $n = 3$ this number equals to 40,320, for $n = 4$ is greater than $2 \cdot 10^{13}$.

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 circuit fanout of each gate output is always equal to 1. As a consequence n^*n reversible circuits can be only built as a cascade of k^*k reversible gates ($k \leq n$).

Definition 3. A set of reversible gates that can be used to build reversible circuits is called a *gate library*.

Many gate libraries have been examined in the literature. The so called *NCT library* for $n \leq 4$ 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 $a_i \in \{0, 1\}$ for $i = 1, 2, 3, 4$, and let x' denotes negation of x .

1^*1 NOT(x_1) gate performs the operation

$$(x_1) \rightarrow (x_1 \oplus 1),$$

2^*2 CNOT(x_1, x_2) gate performs the operation

$$(x_1, x_2) \rightarrow (x_1, (x_1 \oplus a_1) \oplus x_2),$$

3^*3 TOFFOLI(x_1, x_2, x_3) gate performs the operation

$$(x_1, x_2, x_3) \rightarrow (x_1, x_2, (x_1 \oplus a_1)(x_2 \oplus a_2) \oplus x_3),$$

4^*4 TOFFOLI4(x_1, x_2, x_3, x_4) gate performs the operation

$$(x_1, x_2, x_3, x_4) \rightarrow (x_1, x_2, x_3, (x_1 \oplus a_1)(x_2 \oplus a_2)(x_3 \oplus a_3) \oplus x_4),$$

3^*3 FREDKIN(x_1, x_2, x_3) gate performs the operation

$$(x_1, x_2, x_3) \rightarrow (x_1, x_1' x_2 + x_1 x_3, x_1 x_2 + x_1' x_3).$$

The above defined i^*i gates, where $i = 1, 2, 3, 4$, (in short, denoted by N, C, T, T4 and F, respectively) invert input x_i if and only if the values of inputs x_1, x_2, \dots, x_{i-1} differ from corresponding a_1, a_2, \dots, a_{i-1} coefficients, passing these inputs unchanged to corresponding outputs. Signals which are passed unchanged from input to output of the gate are called *control lines*. The signal x_i which can be modified by the gate is called *target*. In the Fredkin gate x_1 is the control line. When its input signal is 0 then the signals x_2 and x_3 are passed to outputs unchanged, and when its input signal is 1 then the signals x_2 and x_3 are swapped.

Some commonly used names for describing control lines have been introduced. If all a_1, a_2, \dots, a_i coefficients equals zero, the gate has *positive-polarity* control lines. Alternatively, if all a_1, a_2, \dots, a_i coefficients equals one, the gate has *negative-polarity* control lines. These names come from 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 is used if all values of the coefficients a_1, a_2, \dots, a_i are allowed to be either 0 or 1. Pictorial symbols of the above defined positive-polarity gates are shown in Fig. 1, and the pictorial symbol of Fredkin gate is shown in Fig. 2. Pictorial symbols of mixed-polarity gates are shown in Fig. 4 (gates B3, B4, B5, B7, B8 and B9). In the description of a gate from NCT library, e.g. T3-210 (see Fig. 1), the first number appearing after the gate's name (in this case: T) denotes the target line (3) while the numbers 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 gate's 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 different positive-polarity gates belonging to NCT library which can appear in 4-line circuits is 32 (4 NOT gates, 12 CNOT gates, 12 Toffoli gates and 4 T4 gates).

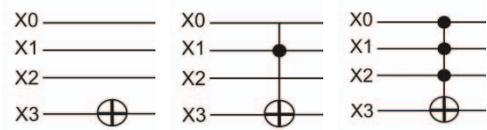


Fig. 1. Graphical representations of reversible gates N3, C3-1 and T3-210.

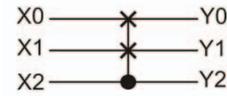


Fig. 2. Graphical representation of Fredkin gate F10-2.

III. RECONFIGURABLE REVERSIBLE GATE

Reconfigurable reversible gate (RRG) replaces one of 32 gates from the NCT library in the cipher. This is why besides 4 inputs transferring information data there have to be five lines for choosing one of 32 types of gates.

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 shown in Fig. 3 has been constructed with the relatively small quantum cost equal to 79. 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 functioning of the gate B6 is determined by signals K. Gates B7, B8 and B9 reconstruct initial constant signals, while gates B10, B11 and B12 fix the order of output signals Y3, Y2, Y1 and Y0. The RRG built from basic reversible gates is shown in Fig. 4.

For simulation of RRG a procedure B_FREDKIN for representing functioning of Fredkin gate and a procedure B_NCT for representing any 4-input gate from NCT library have been constructed in VHDL and shown in Tables I and II.

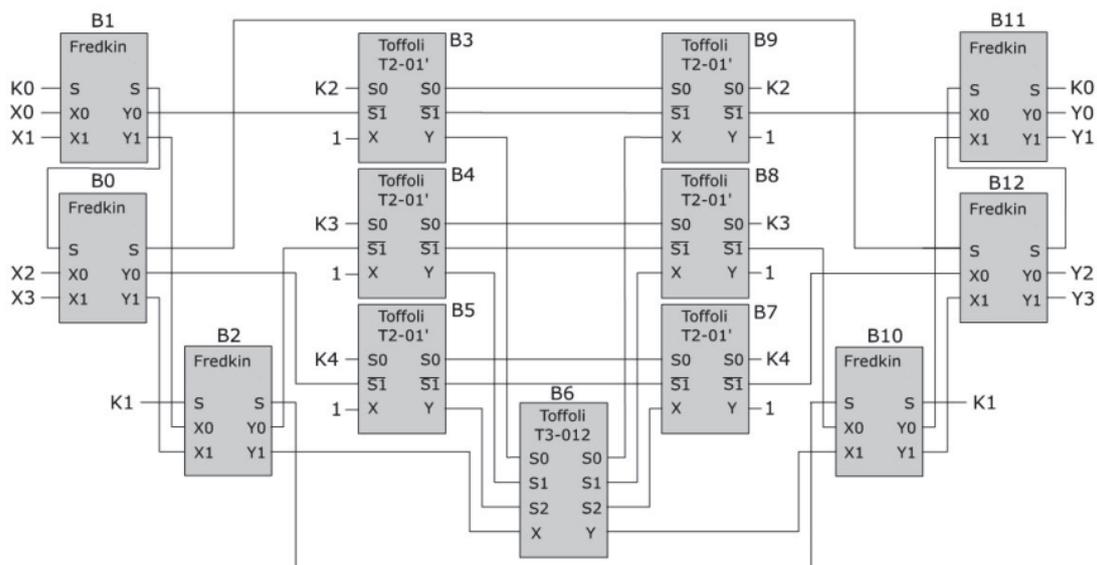


Fig. 3. General scheme of Reconfigurable Reversible Gate (RRG).

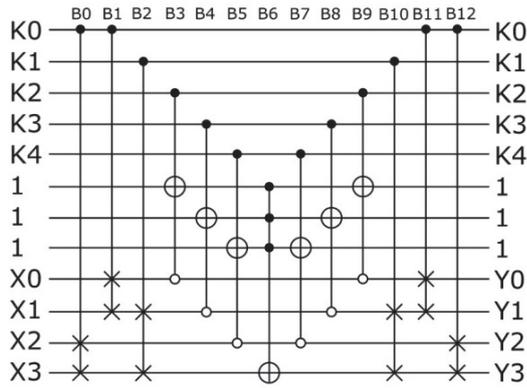


Fig. 4. Reconfigurable Reversible Gate (RRG).

TABLE I. PROCEDURE B_FREDKIN

```

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
is
begin
  if X(C)='0' then Y:=X;
    else Y:=X; Y(L1):=X(L2);
      Y(L2):=X(L1);
    end if;
end procedure B_FREDKIN;

```

TABLE II. PROCEDURE B_NCT

```

procedure B_NCT
(X : in std_logic_vector; -- input line
 C1 : in integer; -- control line
 N1 : in std_logic; -- negative control line C1
 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
 CO : in integer; -- control line output
 Y : out std_logic_vector) -- output line
begin
  if C1=12 then Y:=X; Y(CO):= not X(CO);
  elsif C2=12 then Y:=X;
    Y(CO):= (X(C1) xor N1) xor X(CO);
  elsif C3=12 then Y:=X;
    Y(CO):= ((X(C1) xor N1)
      and (X(C2) xor N2))xor X(CO);
    else Y:=X;
      Y(CO):= ((X(C1) xor N1)
        and (X(C2) xor N2)
        and (X(C3) xor N3))xor X(CO);
    end if;
end procedure B_NCT;

```

In the procedure B_NCT signals C1, C2 and C3 determine number of line to which control inputs of Toffoli gate are attached while the number of controlled line is determined by signal CO. Binary signals N1, N2 and N3 determine polarization of the control signal C_i . Value N1 = 1 indicates that i -th signal is a negative control signal.

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));
DOUT(3 downto 0) <= (KX(13)(11) &
  KX(13)(10) & KX(13)(9) & KX(13)(8));

```

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. The above presented circuit requires from 8 to 30 logical elements in a cascade, depending on the type of FPGA. In our simulation the circuits EP3SL50F484C2 (Stratix III), EP3C5F256C6 (Cyclone III) and EP1K30TC144-3 (ACEX1K) have been used [11].

IV. IMPLEMENTATION OF ENCRYPTION BY RECONFIGURABLE REVERSIBLE GATES

General idea of a cipher built from reversible gates was presented in [9]. It is shown in Fig. 5. The basic element of the cipher is a cascade of 16 4-input reversible gates. The same main key is used for encryption and decryption. The order of gates in the cascade for decryption is reversed in comparison with the cascade for encryption what ensures that it transforms ciphertext into plaintext. It has been proved in [12-13] that optimal circuits for any of the $16!$ (equal to over 2×10^{13}) require at most 15 4-input gates. Thus the circuit shown in Fig. 6 enables realization of any 4-variable reversible function. The circuit has 80 inputs denoted K (with indexes) that are partitioned into groups with 5 inputs in each of them. A 5-line group $K[(5*(i+1) - 1):5*i]$ is used to configure i -th 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 identical outputs.

Detailed description of the main key register as well as the circuit modifying its contents during encryption/decryption are presented in [14].

V. SIMULATION OF CIPHERS

For encryption of 8-bit data two 4-bit ciphers and the main key register were used.

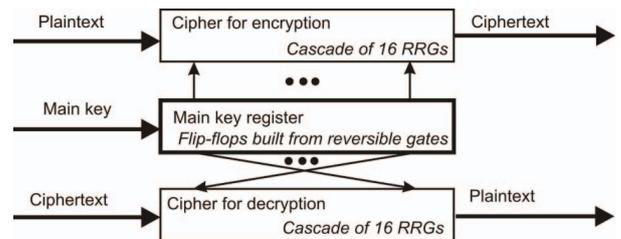


Fig. 5. General idea of a cipher for encryption and decryption built from reversible gates.

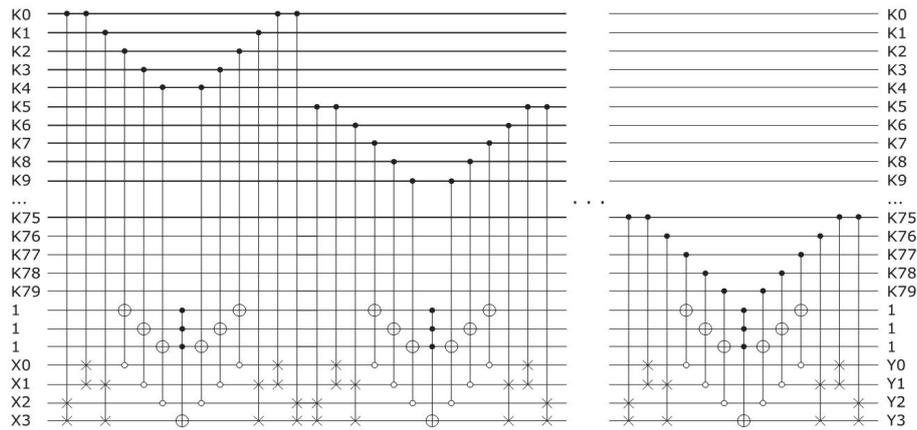


Fig. 6. The cascade of reversible gates implementing the 4-bit cipher.

Results of simulation of a 8-bit cipher using Quartus with module ModelSim are shown in Fig. 7. Simulation of the cipher was performed for subsequent increasing values from 0 to 255 applied to 8 inputs. Realization of the cipher for encryption in FPGA circuit from the series Cyclone IV E manufactured by Altera needed 2042 logical elements LE, while realization of the cipher for decryption needed 1914 LE. The FPGA circuit from the series Cyclone V for such ciphers needed 824 and 748 blocks ALM, respectively.

VI. CONCLUSIONS

The main aim of the 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. The design of RRG is presented for the first time in the literature. An application of this gate to implement ciphers for encryption and decryption of binary data is described. Results of FPGA-based simulation of the cipher built from reversible gates are also presented.

ACKNOWLEDGMENTS

Work leading to this paper was partially supported by the EU COST Action IC 1405 "Reversible Computation – Extending Horizons of Computing".

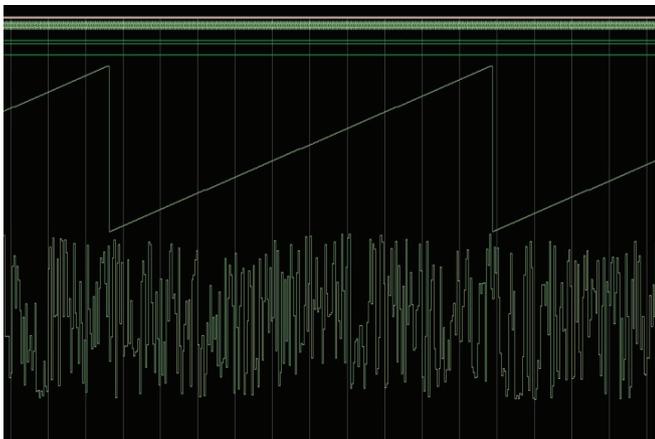


Fig. 7. Simulation results of encryption for a chosen main key.

REFERENCES

- [1] A. De Vos, Reversible Computing. Fundamentals, Quantum Computing, and Applications. Wiley-VCH, Berlin 2010.
- [2] H. Thapliyal and M. Zvolinski, "Reversible logic to cryptographic hardware: a new paradigm," Proc. 49th International Midwest Conference on Circuits and Systems, s. 342-346, 2006.
- [3] N. M. Nayeem, L. Jamal, and H. M. H. Babu, "Efficient reversible Montgomery multiplier and its application to hardware cryptography," Journal of Computer Science, vol. 5, no. 1, pp. 49-56, 2009.
- [4] Y. Zhang, Z. Guan, and Z. Nie, "Function modular design of the DES encryption system based on reversible logic gates," Proc. International Conference on Multimedia Communications, pp. 104-107, 2010.
- [5] A. Banerjee, "Reversible cryptographic hardware with optimized quantum cost and delay," Proc. Annual IEEE India Conference, pp. 1-4, 2010.
- [6] K. Datta and I. Sengupta, "Applications of reversible logic in cryptography and coding theory (Tutorial)," Proc. Conference on VLSI Design (VLSID), 2013.
- [7] K. Datta, V. Shrivastav, I. Sengupta and H. Rahaman, "Reversible logic implementation of AES algorithm," Proc. 8th International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS), pp. 140-144, 2013.
- [8] A.C. Nuthan, C. Nagaraj and V.B. Havyas, "Implementation of Data Encryption Standard Using Reversible Gate Logic," International Journal of Soft Computing and Engineering, vol. 3, no. 3, pp. 270-272, 2013.
- [9] A. Skorupski, M. Pawłowski, K. Gracki, and P. Kerntopf, „FPGA-based modeling of encryption systems implemented in reversible logic” (in Polish), Pomiary Automatyka Kontrola, vol. 58, no. 7, pp. 620-622, 2012.
- [10] A. Skorupski, M. Pawłowski, K. Gracki, and P. Kerntopf, „Reconfiguration of reversible functions using modeling of gates in FPGA” (in Polish), Pomiary Automatyka Kontrola, vol. 60, no. 9, pp. 471-473, 2014.
- [11] M. Pawłowski and A. Skorupski, Design of Complex Digital Devices (in Polish), WKŁ, Warsaw 2010.
- [12] O. Golubitsky and D. Maslov, "A study of optimal 4-bit reversible Toffoli circuits and their synthesis," IEEE Transactions on Computers, vol. 61, no. 9, s. 1341-1353, 2012.
- [13] M. Szyprowski and P. Kerntopf, "A Study of Optimal 4-bit Reversible Circuit Synthesis from Mixed-Polarity Toffoli Gates," Proc. 12th IEEE Conference on Nanotechnology, 2012.
- [14] M. Bryk, "Cipher built from reversible gates" (in Polish), MSc thesis, Institute of Computer Science, Warsaw University of Technology, February 2016.