Observations on the Quantum Circuit of the SBox of AES

Jian Zou[1,2], Yongyang Liu[1,2], Chen Dong[1,2], Wenling Wu[3], Le Dong[4]

[1]Mathematics and Computer Science of Fuzhou University, Fuzhou, China, 350108
[2]Key Lab of Information Security of Network Systems (Fuzhou University), Fuzhou, China, 350108
[3] Institute of Software, Chinese Academy of Sciences, Beijing 100190, P.R. China
[4] Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control, Henan Normal University, Xinxiang, China, 453007
fzuzoujian15@163.com

Abstract. In this paper, we propose some improved quantum circuits to implement the Sbox of AES. Our improved quantum circuits are based on the following strategies. First, we try to find the minimum set of the intermediate variables that can be used to compute the 8-bit output of the Sbox. Second, we check whether some wires store intermediate variables and remain idle until the end. And we can reduce the number of qubit by reusing some certain wires. Third, we try to compute the output of the Sbox without ancillas qubits, because we do not need to be clean up the wires storing the output of the Sbox. This operation will reduce the number of Toffoli gates. Our first quantum circuit only needs 26 qubits and 46 Toffoli gates, while quantum circuit proposed by Langenberg *et al.* required 32 qubits and 55 Toffoli gates. Furthermore, we can also construct our second quantum circuit with 22 qubits and 60 Toffoli gates.

Key words: quantum circuit, AES, Sbox, Grovers algorithm

1 Introduction

Post-quantum cryptography studies the security of cryptographic systems against quantum attackers. Due to the rapid development of quantum computer, many cryptographic schemes have been found out to be insecure in quantum computing. Asymmetric cryptographic primitives encounter devastating attacks due to Shor's algorithm [1]. In contrast to asymmetric cryptographic, the impact of quantum computing on secret-key cryptography is is not so clear. It's well known that Grover's algorithm [2] will solve the problem of finding keys with quadratic speed-up, i.e. $O(2^{n/2})$. It is worth realizing such attack so as to obtain the precise resource estimate for implementing Grover's algorithm.

There are some research on how to implement quantum circuits of AES and its Sbox. In [3], Grassl *et al.* proposed a quantum circuit for the Sbox of AES with 40 qubits, 512 Toffoli, 469 CNOT, and 4 NOT gates. In addition, they [3] also proposed a quantum circuit for Sbox with 9 qubits, 1385 Toffoli plus 1551 CNOT or NOT gates. Compare with their first construction, this circuit should need more Toffoli gates so as to use only one ancilla qubit. Almazrooie *et al.* in [4] also presented a quantum circuit for the Sbox with 56 qubits and 448 Toffoli gates. In [5], Kim *et al.* presented an improved quantum circuit for the Sbox with 40 qubits and 448 Toffoli gates. Saravanan and Kalpana in [6] proposed a quantum circuit for Sbox with 35 Toffoli, 152 CNOT, and 4 NOT gates, which required dozens of garbage outputs qubits. By utilizing the algebraic structure of the Sbox [7], Langenberg *et al.* [8] proposed a quantum circuit for Sbox with 32 qubits, 55 Toffoli, 314 CNOT, and 4 NOT gates.

In this paper, we try to construct some improved quantum circuits for the Sbox of AES. Since the cost of Toffoli gate is more expensive than the gates in Clifford group, our primary objective is to reduce the number of qubit and Toffoli gates in this paper. Our results are summarized in Table 1.

Table 1. Comparison of circuit designs for the Sbox of AES

Number of qubits	Number of Toffoli gate	Source
40	512	[3]
9	1385	[3]
56	448	[4]
40	448	[5]
32	55	[8]
26	46	Section 3
22	60	Appendix B

Organization. This paper is organized as follows. In Section 2, we make a brief introduction to the structure of the Sbox of AES. Section 3 show our new techniques for constructing the improved quantum circuits for Sbox. Section 4 concludes this paper.

2 AES algorithm

The round function of AES consists of the following four operations:

1. AddRoundKey: The AddRoundKey operation xor the round key to the state.
2. SubBytes: The SubBytes transformation applies the Sbox operation to each 8-bit cell of the state.
3. ShiftRows: The ShiftRows transformation cyclically rotates the cells of the i-th row leftward by shift vector.
4. MixColumns: In the MixColumns operation, each column of the state is multiplied by an MDS matrix.

Since we just consider how to obtain some improved quantum circuits for the Sbox of AES, we just omit the left three operations of AES. For a full description of AES, please refer to [9].

2.1 The Sbox of AES

There are several ways to implement Sbox. On the one hand, we can implement Sbox as a look-up table. On the other hand, if we treat an input byte as an element $b \in GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$, then the 8-bit output of Sbox $(s_0, s_1, \cdots, s_7)$ can be realized by computing multiplicative inverse of b followed by affine transformations. Define b' as b^{-1}, then $(s_0, s_1, \cdots, s_7)^T = M \cdot b'^T + C^T$, where $C^T = [1, 1, 0, 0, 0, 1, 1, 0]$ and

$$M = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{pmatrix} \tag{1}$$

2.2 The algebraic structure of the Sbox of AES

In [7], Boyar and Peralta proposed a low-depth circuit for the Sbox in AES with only 34 AND gates. It is easy to construct a quantum circuit with 68 Toffoli gates combining Bennett's method [10] with Boyar and Peralta's work. In [8], Langenberg *et al.* constructed their quantum circuit by utilizing the algebraic structure of the Sbox proposed by Boyar and Peralta [7]. In detail, their proposed a quantum circuit of the of Sbox AES with 32 qubits, 55 Toffoli, 314 CNOT, and 4 NOT gates. Since we try to improve their quantum circuit of the Sbox, we also focus on the work proposed by Boyar and Peralta [7]. However, our techniques can also be applied to other constructions of the Sbox of AES, i.e. the other work proposed by Boyar and Peralta [11].

In [7], Boyar and Peralta observed the Sbox of AES could be represented as $S(x) = B \cdot F(U \cdot x)$, where matrices $B \in F_2^{8\times 18}$, $U \in F_2^{22\times 8}$, and $F : F_2^{22} \to F_2^{18}$ is a non-linear function. The B, U and F are presented as follows.

The matrix $B \in F_2^{8\times 18}$ takes $x_0, x_1, \cdots, x_7$ as input, and outputs $x_7, y_1, \cdots, y_{21}$, which are inputs to the non-linear function F.

$$y_{14}=x_3+x_5,\ \ y_{13}=x_0+x_6,\ \ y_9=x_0+x_3,\ \ y_8=x_0+x_5,\ \ t_0=x_1+x_2,$$
$$y_1=t_0+x_7,\ \ y_4=y_1+x_3,\ \ y_{12}=y_{13}+y_{14},\ \ y_2=y_1+x_0,\ \ y_5=y_1+x_6,$$
$$y_3=y_5+y_8,\ \ t_1=x_4+y_{12},\ \ y_{15}=t_1+x_5,\ \ y_{20}=t_1+x_1,\ \ y_6=y_{15}+x_7,$$
$$y_{10}=y_{15}+t_0,\ \ y_{11}=y_{20}+y_9,\ \ y_7=x_7+y_{11},\ \ y_{17}=y_{10}+y_{11},$$
$$y_{19}=y_{10}+y_8,\ \ y_{16}=t_0+y_{11},\ \ y_{21}=y_{13}+y_{16},\ \ y_{18}=x_0+y_{16}.$$

The non-linear function $F: F_2^{22} \rightarrow F_2^{18}$ takes $x_0, x_1, \cdots, x_7$ as input, and outputs $z_0, z_1, \cdots, z_{17}$, which are inputs to the matrix U.

$$t_2=y_{12}\cdot y_{15},\ \ t_3=y_3\cdot y_6,\ \ t_4=t_3+t_2,\ \ t_5=y_4\cdot x_7,\ \ t_6=t_5+t_2,$$
$$t_7=y_{13}\cdot y_{16}\ \ t_8=y_5\cdot y_1,\ \ t_9=t_8+t_7,\ \ t_{10}=y_2\cdot y_7\ \ t_{11}=t_{10}+t_7,$$
$$t_{12}=y_9\cdot y_{11},\ \ t_{13}=y_{14}\cdot y_{17}\ \ t_{14}=t_{13}+t_{12},\ \ t_{15}=y_8\cdot y_{10},\ \ t_{16}=t_{15}\cdot t_{12}$$
$$t_{17}=t_4\cdot t_{14},\ \ t_{18}=t_6+t_{16},\ \ t_{19}=t_9+t_{14},\ \ t_{20}=t_{11}+t_{16},\ \ t_{21}=t_{17}+y_{20},$$
$$t_{22}=t_{18}+y_{19}\ \ t_{23}=t_{19}+y_{21},\ \ t_{24}=t_{20}+y_{18}\ \ t_{25}=t_{21}+t_{22},$$
$$t_{26}=t_{21}\cdot t_{23},\ \ t_{27}=t_{24}+t_{26},\ \ t_{28}=t_{25}\cdot t_{27},\ \ t_{29}=t_{28}+t_{22},\ \ t_{30}=t_{23}+t_{24},$$
$$t_{31}=t_{22}+t_{26},\ \ t_{32}=t_{31}\cdot t_{30},\ \ t_{33}=t_{32}+t_{24},\ \ t_{34}=t_{23}+t_{33},$$
$$t_{35}=t_{27}+t_{33},\ \ t_{36}=t_{24}\cdot t_{35}\ \ t_{37}=t_{36}+t_{34},\ \ t_{38}=t_{27}+t_{36},\ \ t_{39}=t_{29}\cdot t_{38},$$
$$t_{40}=t_{25}+t_{39},\ \ t_{41}=t_{40}+t_{37},\ \ t_{42}=t_{29}+t_{33},\ \ t_{43}=t_{29}+t_{40},$$
$$t_{44}=t_{33}+t_{37},\ \ t_{45}=t_{42}+t_{41},\ \ z_0=t_{44}\cdot y_{15},\ \ z_1=t_{37}\cdot y_6,\ \ z_2=t_{33}\cdot x_7,$$
$$z_3=t_{43}\cdot y_{16},\ \ z_4=t_{40}\cdot y_1,\ \ z_5=t_{29}\cdot y_7,\ \ z_6=t_{42}\cdot y_{11},\ \ z_7=t_{45}\cdot y_{17},$$
$$z_8=t_{41}\cdot y_{10},\ \ z_9=t_{44}\cdot y_{12},\ \ z_{10}=t_{37}\cdot y_3,\ \ z_{11}=t_{33}\cdot y_4,\ \ z_{12}=t_{43}\cdot y_{13},$$
$$z_{13}=t_{40}\cdot y_5,\ \ z_{14}=t_{29}\cdot y_2,\ \ z_{15}=t_{42}\cdot y_9,\ \ z_{16}=t_{45}\cdot y_{14},\ \ z_{17}=t_{41}\cdot y_8.$$

The inputs to the matrix U are $z_0, z_1, \cdots, z_{17}$, while the outputs are $s_0, s_1, \cdots, s_7$.

$$t_{46}=z_{15}+z_{16},\ \ t_{47}=z_{10}+z_{11},\ \ t_{48}=z_5+z_{13},\ \ t_{49}=z_9+z_{10},$$
$$t_{50}=z_2+z_{12},\ \ t_{51}=z_2+z_5,\ \ t_{52}=z_7+z_8,\ \ t_{53}=z_0+z_3,\ \ t_{54}=z_6+z_7,$$
$$t_{55}=z_{16}+z_{17},\ \ t_{56}=z_{12}+t_{48},\ \ t_{57}=t_{50}+t_{53},\ \ t_{58}=z_4+t_{46},$$
$$t_{59}=z_3+t_{54},\ \ t_{60}=t_{46}+t_{57},\ \ t_{61}=z_{14}+t_{57},\ \ t_{62}=t_{52}+t_{58},$$
$$t_{63}=t_{49}+t_{58},\ \ t_{64}=z_4+t_{59}, t_{65}=t_{61}+t_{62}, t_{66}=z_1+t_{63},\ \ s_0=t_{59}+t_{63},$$
$$s_6=t_{56}\ XNOR\ t_{62},\ \ s_7=t_{48}\ XNOR\ t_{60},\ \ t_67=t_{64}+t_{65},\ \ s_3=t_{53}+t_{66},$$
$$s_4=t_{51}+t_{66},\ \ s_5=t_{47}+t_{65},\ \ s_1=t_{64}\ XNOR\ s_3,\ \ s_2=t_{55}\ XNOR\ t_{67}.$$

3 Main Result

In this article, we try to improve the quantum circuit proposed by Langenberg *et al.* in [8]. In detail, we propose two improved quantum circuits for the Sbox of AES. The goal of our first quantum circuit is reducing the number of Toffoli gates as small as possible, while we try to construct our second quantum circuit with the least number of qubits. Compared with the quantum circuit in [8], our two quantum circuits not only reduce the number of Toffoli gates, but also reduce the number of qubits. In the following, we will show how to construct our first quantum circuit. Since our second quantum circuit is similar to our first quantum circuit, we just show the detail of our first quantum circuit in this section. The detail of our second quantum circuit is shown in Appendix B. As shown in Appendix A and B, our quantum circuits adopt the same notation in [8]. The 8-bit input of Sbox and the 8-bit output of Sbox are expressed as $U[0], \cdots, U[7]$ and $s[0], \cdots, s[7]$ respectively. We also use T (ancillas qubits) to store the intermediate values of computation, which shall return to zero at the end of computation. Note that we do not need the ancilla qubit Z in our quantum circuit.

Our first quantum circuit is constructed by adopting the following strategies.

1. In the quantum circuit, we shall clean up the wires with the intermediate values, while the wires of the output of Sbox do not need to be clean up. In order to reduce the number of Toffoli gates, we shall apply Toffoli gates to the wires of the output of the Sbox.
2. In the quantum circuit proposed by Langenberg *et al.* [8], some wires remained idle until the end of the quantum circuit. By uncomputing these wires, we can reuse these wires so as to reduce the number of qubits.

Note that Langenberg *et al.* [8] also used the above strategies to construct their quantum circuit for the Sbox of AES. However, we can improve their quantum circuit with the following observations, which utilizes the linear relationship between different variables.

Observation 1. As pointed out in [7], the 18 values of $z_0, \cdots, z_{17}$ can be obtained with the knowledge of $t_{29}, t_{33}, t_{37}, t_{40}, t_{42}, t_{42}, t_{43}, t_{44}, t_{45}$ and $x_7, y_0, \cdots, y_{17}$, where $y_0, \cdots, y_{17}$ can be obtained by the linear combination of $x_0, x_1, \cdots, x_7$. In addition, t_{29}, t_{33}, t_{37}, t_{40}, t_{41}, t_{42}, t_{43}, t_{44}, t_{45} can be obtained by the linear combination of $t_{29}, t_{33}, t_{37}, t_{40}$. In other words, we can obtain $z_0, \cdots, z_{17}$ with the knowledge of $t_{29}, t_{33}, t_{37}, t_{40}$ and $x_0, x_1, \cdots, x_7$.

According to Observation 1, we only need to store $t_{29}, t_{33}, t_{37}, t_{40}$ and x_0, x_1, $\cdots$, x_7 instead of t_{29}, t_{33}, t_{37}, t_{40}, t_{42}, t_{42}, t_{43}, t_{44}, t_{45} and $x_7, y_0, \cdots, y_{17}$, which could save some qubits.

Observation 2. As pointed out in[7], the 8-bit output of Sbox $s_0, s_1, \cdots, s_7$ can be seen as a linear combination of the 18 values of $z_0, \cdots, z_{17}$. Given the 18 values of $z_0, \cdots, z_{17}$, we can express the linear expression of s_i (for $0 \leq i \leq 7$) as follows.

$$s_0 = z_3 + z_4 + z_6 + z_7 + z_9 + z_{10} + z_{15} + z_{16},$$

$$s_1 = \overline{z_0 + z_1 + z_6 + z_7 + z_9 + z_{10} + z_{15} + z_{16}},$$

$$s_2 = \overline{z_0 + z_2 + z_6 + z_8 + z_{12} + z_{14} + z_{15} + z_{17}},$$

$$s_3 = z_0 + z_1 + z_3 + z_4 + z_9 + z_{10} + z_{15} + z_{16},$$

$$s_4 = z_1 + z_2 + z_4 + z_5 + z_9 + z_{10} + z_{15} + z_{16},$$

$$s_5 = z_0 + z_2 + z_3 + z_4 + z_7 + z_8 + z_{10} + z_{11} + z_{12} + z_{14} + z_{15} + z_{16},$$

$$s_6 = \overline{z_4 + z_5 + z_7 + z_8 + z_{12} + z_{13} + z_{15} + z_{16}},$$

$$s_7 = \overline{z_0 + z_2 + z_3 + z_5 + z_{12} + z_{13} + z_{15} + z_{16}},$$

where $'+'$ means $\oplus$, and $\bar{s}$ applies the NOT operation on s.

As shown in [8], the quantum circuit proposed by Langenberg *et al.* used 15 ancilla qubits to store the intermediate values, which could be used to compute $z_0, \cdots, z_{17}$. Based on our observation 1 and 2, we can reduce the number of Toffoli gates and qubits of the quantum circuits in [8] simultaneously as follows. According to our **Observation 1**, we do not need to store the values of t_{41}, t_{42}, t_{43}, t_{44}, t_{45}, which saves 5 qubits. In other words, we only need 10 ancilla qubits to store the intermediate values that could be used to compute $z_0, \cdots, z_{17}$. According to our **Observation 2**, we observe that we could compute the 8-bit output of the Sbox without utilizing the ancilla qubit Z in our quantum circuit. This observation can reduce the number of Toffoli gate and the number of qubit further, because we do not need to recompute the toffoli gates so as to initialize ancilla qubit Z in our quantum circuit. Given the values of $z_0, \cdots, z_{17}$, we will show the detail of how to obtain the 8-bit output of Sbox without the ancilla qubit Z in the following. Compared with the quantum circuits proposed by Langenberg *et al.* [8], our quantum circuit only needs 26 qubits, 46 Toffoli, 304 CNOT, and 4 NOT gates.

Algorithm 1 Compute the output of Sbox without the ancilla qubit Z

Require:

input, $z_0, \cdots, z_{17}$;

input, $s_0 = 0, \cdots, s_7 = 0$;

$s_2 = z_{12}$; $s_6 = s_2$;
$s_2 = z_{14}$; $s_5 = s_2$;
$s_4 = z_5$; $s_6 = s_4$;
$s_1 = z_1$; $s_3 = s_1$;$s_4 = s_1$;
$s_7 = z_8$; $s_4 = s_7$;$s_6 = s_7$;
$s_7 = z_2$; $s_1 = s_7$;$s_3 = s_7$;$s_4 = s_7$;
$s_7 = z_0$; $s_1 = s_7$;$s_2 = s_7$;$s_3 = s_7$;$s_5 = s_7$;
$s_6 = z_{13}$; $s_7 = s_6$;
$s_0 = z_3$; $s_4 = s_0$;$s_6 = s_0$;$s_7 = s_0$;
$s_0 = z_4$; $s_1 = s_0$;$s_2 = s_0$;$s_3 = s_0$;$s_4 = s_0$;$s_5 = s_0$;$s_6 = s_0$;
$s_0 = z_6$; $s_2 = s_0$;$s_5 = s_0$;$s_6 = s_0$;
$s_0 = z_7$; $s_3 = s_0$;$s_4 = s_0$;$s_5 = s_0$;$s_6 = s_0$;
$s_0 = z_9$; $s_5 = s_0$;
$s_0 = z_{10}$; $s_6 = s_0$;$s_7 = s_0$;
$s_0 = z_{16}$; $s_2 = s_0$;
$s_0 = z_{15}$; $s_1 = s_0$;$s_2 = s_0$;$s_3 = s_0$;$s_4 = s_0$;$s_5 = s_0$;$s_6 = s_0$;$s_7 = s_0$;
$s_5 = z_{11}$;
$s_2 = z_{17}$;
Output s_0, $\overline{s_1}$, $\overline{s_2}$, s_3, s_4, s_5, $\overline{s_6}$, $\overline{s_7}$ as the 8-bit output of the Sbox;

Based on the above two observations, we can construct our first quantum circuit with 26 qubits, 46 Toffoli, 304 CNOT, and 4 NOT gates. We will show a detailed description of this quantum circuit in Appendix A.

4 Conclusion

By using our two observations, we can improve the quantum circuits for Sbox proposed by Langenberg *et al.* [8]. Note that our observations can also be applied to the other constructions of the Sbox of AES [11]. With our quantum circuits for Sbox, we can improve the quantum circuits for AES with the techniques in [8], such as parallelization and "zig-zag" method.

5 References

[1] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput., 26(5):1484C1509, October 1997.

[2] Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In Proceedings of the 28th Annual ACM Symposium on the Theory of Computing STOC 1996, pages 212C219, 1996.

[3] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. Applying Grovers Algorithm to AES: Quantum Resource Estimates. In Tsuyoshi Takagi, editor, Post-Quantum Cryptography PQCrypto 2016, volume 9606 of Lecture Notes in Computer Science, pages 29C43. Springer, 2016.

[4] Mishal Almazrooie, Azman Samsudin, Rosni Abdullah, and Kussay N. Mutter. Quantum reversible circuit of AES-128. Quantum Information Processing, 17(5):112, 2018.

[5] Panjin Kim, Daewan Han, and Kyung Chul Jeong. TimeCspace complexity of quantum search algorithms in symmetric cryptanalysis: applying to AES and SHA-2. Quantum Information Processing, 17:339, 2018.

[6] P. Saravanan and P. Kalpana. Novel Reversible Design of Advanced Encryption Standard Cryptographic Algorithm forWireless Sensor Networks. Wireless Personal Communications, 100(4):1427C1458, 2018.

[7] Joan Boyar and Rene Peralta. A New Combinational Logic Minimization Technique with Applications to Cryptology. In Paola Festa, editor, International Symposium on Experimental Algorithms SEA 2010, volume 6049 of Lecture Notes in Computer Science, pages 178C189. Springer, 2010. Preprint available at https://eprint.iacr.org/2009/191.

[8] Brandon Langenberg, Hai Pham, and Rainer Steinwandt. Reducing the cost of implementing AES as a quantum circuit. Cryptology ePrint Archive, Report 2019/854, 2019.

[9] NIST. Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, November 2001.

[10] Charles H. Bennett. Logical Reversibility of Computation. IBM Journal of Research and Development, 17(6):525C532, 1973.

[11] Joan Boyar and Rene Peralta. A depth-16 circuit for the AES S-box. Cryptology ePrint Archive: Report 2011/332, June 2011. Available at https://eprint.iacr.org/2011/332.

6 Appendix A

In this section, we show the detail of our quantum circuit with 26 qubits, 46 Toffoli, 304 CNOT, and 4 NOT gates.

```
import math
from projectq.ops import CNOT, Measure, X, Toffoli
from projectq import MainEngine
from projectq.meta import Compute , Uncompute
from projectq.backends import CircuitDrawer, ResourceCounter,
    ClassicalSimulator
import projectq.libs.math
drawing_engine = CircuitDrawer()
resource_counter = ResourceCounter()
sim = ClassicalSimulator()
eng = MainEngine(sim)
def aes box (eng) :
    U = eng . allocate qureg (8)
    T = eng . allocate qureg (10)
    S = eng . allocate qureg (8)
    input_m = [0]*(8)
    output_m = [0]*(8)

    with Compute ( eng ) :
        CNOT | (U[0] ,U[5])
        CNOT | (U[3] ,U[5])
        CNOT | (U[6] ,U[5])
        CNOT | (U[0] ,U[4])
        CNOT | (U[3] ,U[4])
        CNOT | (U[6] ,U[4])
        Toffoli | (U[5] ,U[4] ,T[0])        #t2
        CNOT | (T[0] ,T[5])

        CNOT | (U[1] ,U[3])
        CNOT | (U[2] ,U[3])
        CNOT | (U[7] ,U[3])
```

```
Toffoli | (U[3] ,U[7] ,T[0])        #t6

CNOT | (U[0] ,U[6])
CNOT | (U[0] ,U[2])
CNOT | (U[4] ,U[2])
CNOT | (U[5] ,U[2])
CNOT | (U[6] ,U[2])
Toffoli | (U[6] ,U[2] ,T[1])        #t7
CNOT | (T[1] ,T[2])

CNOT | (U[2] ,U[1])
CNOT | (U[4] ,U[1])
CNOT | (U[5] ,U[1])
CNOT | (U[7] ,U[1])
CNOT | (U[1] ,U[0])
CNOT | (U[6] ,U[0])
Toffoli | (U[1] ,U[0] ,T[1])        #t9

CNOT | (U[1] ,U[6])
CNOT | (U[0] ,U[2])
Toffoli | (U[6] ,U[2] ,T[2])        #t11

CNOT | (U[6] ,U[3])
CNOT | (U[7] ,U[2])
Toffoli | (U[3] ,U[2] ,T[3])        #t12
CNOT | (T[3] ,T[4])

CNOT | (U[1] ,U[6])
CNOT | (U[5] ,U[6])
CNOT | (U[2] ,U[0])
CNOT | (U[4] ,U[0])
CNOT | (U[7] ,U[0])
Toffoli | (U[6] ,U[0] ,T[3])        #t14

CNOT | (U[6] ,U[3])
CNOT | (U[2] ,U[0])
Toffoli | (U[3] ,U[0] ,T[4])        #t16

CNOT | (T[3] ,T[1])        #t19

CNOT | (U[1] ,U[3])
CNOT | (U[7] ,U[4])
Toffoli | (U[3] ,U[4] ,T[5])        #t4
```

```
CNOT | (T[5] ,T[3])        #t17

CNOT | (T[4] ,T[0])        #t18

CNOT | (T[2] ,T[4])        #t20

CNOT | (U[1] ,U[6])
CNOT | (U[2] ,U[6])
CNOT | (U[3] ,U[6])
CNOT | (U[6] ,T[3])        #t21

CNOT | (U[0] ,U[1])
CNOT | (U[3] ,U[1])
CNOT | (U[1] ,T[0])        #t22

CNOT | (U[1] ,U[5])
CNOT | (U[4] ,U[5])
CNOT | (U[6] ,U[5])
CNOT | (U[7] ,U[5])
CNOT | (U[5] ,T[1])        #t23

CNOT | (U[1] ,U[4])
CNOT | (U[3] ,U[4])
CNOT | (U[5] ,U[4])
CNOT | (U[4] ,T[4])        #t24

Toffoli | (T[3] ,T[1] ,T[6])        #t26
CNOT | (T[0] ,T[3])        #t25

CNOT | (T[4] ,T[7])
CNOT | (T[6] ,T[7])        #t27

CNOT | (T[0] ,T[6])        #t31
Toffoli | (T[3] ,T[7] ,T[0])        #t29

CNOT | (T[1] ,T[4])        #t30

Toffoli | (T[6] ,T[4] ,T[9])        #t32

CNOT | (T[1] ,T[4])
#T[4] is set to t24 again

CNOT | (T[4] ,T[9])        #t33
CNOT | (T[9] ,T[1])        #t34
```

```
CNOT | (T[9] ,T[7])          #t35

Toffoli | (T[4] ,T[7] ,T[8])         #t36

CNOT | (T[9] ,T[7])
#T[7] is set to t27 again

CNOT | (T[8] ,T[1])         #t37
CNOT | (T[8] ,T[7])         #t38

Toffoli | (T[0] ,T[7] ,T[3])         #t40

# The T[0-9] are assigned as follows. T[0]=t29, T[1]=t37, T[2]=t11,
T[3]=t40, T[4]=t24, T[5]=t4, T[6]=t31, T[7]=t38, T[8]=t36, T[9]=t33.

CNOT | (U[0] ,U[2])
CNOT | (U[1] ,U[2])
CNOT | (U[6] ,U[2])          # for z16

CNOT | (U[1] ,U[4])
CNOT | (U[3] ,U[4])
CNOT | (U[5] ,U[4])         # for z1

CNOT | (U[1] ,U[6])
CNOT | (U[3] ,U[6])
CNOT | (U[4] ,U[6])
CNOT | (U[5] ,U[6])
CNOT | (U[7] ,U[6])         # for z11

CNOT | (U[1] ,U[0])
CNOT | (U[3] ,U[0])         # for z13

CNOT | (U[0] ,U[3])
CNOT | (U[2] ,U[3])
CNOT | (U[6] ,U[3])         # for z14

# The U[0-7] are assigned as follows. U[0]=y5, U[1]=y19, U[2]=y14,
U[3]=y2, U[4]=y6, U[5]=y21, U[6]=y4, U[7]=x7.
```

```
CNOT | (U[0], U[3])
CNOT | (T[0], T[3])
Toffoli | (T[3], U[3], S[2])         #z12
CNOT | (S[2], S[6])
CNOT | (U[0], U[3])
```

CNOT | (T[0], T[3])
Toffoli | (T[0], U[3], S[2]) #z14
CNOT | (S[2], S[5])

CNOT | (U[0], U[6])
CNOT | (U[1], U[6])
CNOT | (U[2], U[6])
CNOT | (U[4], U[6])
CNOT | (U[5], U[6])
Toffoli | (T[0], U[6], S[4]) #z5
CNOT | (S[4], S[6])
CNOT | (U[0], U[6])
CNOT | (U[1], U[6])
CNOT | (U[2], U[6])
CNOT | (U[4], U[6])
CNOT | (U[5], U[6])

Toffoli | (T[1], U[4], S[1]) #z1
CNOT | (S[1], S[3])
CNOT | (S[1], S[4])

CNOT | (U[1], U[6])
CNOT | (U[2], U[6])
CNOT | (U[3], U[6])
CNOT | (T[1], T[3])
Toffoli | (T[3], U[6], S[7]) #z8
CNOT | (S[7], S[4])
CNOT | (S[7], S[6])
CNOT | (U[1], U[6])
CNOT | (U[2], U[6])
CNOT | (U[3], U[6])
CNOT | (T[1], T[3])

Toffoli | (T[9], U[7], S[7]) #z2
CNOT | (S[7], S[1])
CNOT | (S[7], S[3])
CNOT | (S[7], S[4])

CNOT | (U[7], U[4])
CNOT | (T[9], T[1])

```
Toffoli | (T[1], U[4], S[7])        #z0
CNOT | (S[7] ,S [1])
CNOT | (S[7] ,S [2])
CNOT | (S[7] ,S [3])
CNOT | (S[7] ,S [5])
CNOT | (U[7] ,U[4])
CNOT | (T[9], T[1])


Toffoli | (T[3], U[0], S[6])        #z13
CNOT | (S[6], S[7])


CNOT | (U[0] ,U[5])
CNOT | (U[3] ,U[5])
CNOT | (T[0], T[3])
Toffoli | (T[3], U[5], S[0])        #z3
CNOT | (S[0], S[4])
CNOT | (S[0], S[6])
CNOT | (S[0], S[7])
CNOT | (U[0] ,U[5])
CNOT | (U[3] ,U[5])
CNOT | (T[0], T[3])


CNOT | (U[1] ,U[6])
CNOT | (U[2] ,U[6])
CNOT | (U[3] ,U[6])
CNOT | (U[4] ,U[6])
Toffoli | (T[3], U[6], S[0])        #z4
CNOT | (S[0], S[1])
CNOT | (S[0], S[2])
CNOT | (S[0], S[3])
CNOT | (S[0], S[4])
CNOT | (S[0], S[5])
CNOT | (S[0], S[6])
CNOT | (U[1] ,U[6])
CNOT | (U[2] ,U[6])
CNOT | (U[3] ,U[6])
CNOT | (U[4] ,U[6])


CNOT | (U[0], U[7])
CNOT | (U[1], U[7])
CNOT | (U[2], U[7])
```

CNOT | (U[4], U[7])
CNOT | (U[5], U[7])
CNOT | (U[6], U[7])
CNOT | (T[0], T[9])
Toffoli | (T[9], U[7], S[0]) #z6
CNOT | (S[0], S[2])
CNOT | (S[0], S[5])
CNOT | (S[0], S[6])
CNOT | (U[0], U[7])
CNOT | (U[1], U[7])
CNOT | (U[2], U[7])
CNOT | (U[4], U[7])
CNOT | (U[5], U[7])
CNOT | (U[6], U[7])
CNOT | (T[0], T[9])

CNOT | (U[0] ,U[7])
CNOT | (U[3] ,U[7])
CNOT | (U[4] ,U[7])
CNOT | (U[5] ,U[7])
CNOT | (T[0], T[9])
CNOT | (T[3], T[9])
CNOT | (T[1], T[9])
Toffoli | (T[9], U[7], S[0]) #z7
CNOT | (S[0], S[3])
CNOT | (S[0], S[4])
CNOT | (S[0], S[5])
CNOT | (S[0], S[6])
CNOT | (U[0] ,U[7])
CNOT | (U[3] ,U[7])
CNOT | (U[4] ,U[7])
CNOT | (U[5] ,U[7])
CNOT | (T[0], T[9])
CNOT | (T[3], T[9])
CNOT | (T[1], T[9])

CNOT | (U[0] ,U[3])
CNOT | (U[2] ,U[3])
CNOT | (T[1], T[9])
Toffoli | (T[9], U[3], S[0]) #z9
CNOT | (S[0], S[5])
CNOT | (U[0] ,U[3])
CNOT | (U[2] ,U[3])

CNOT | (T[1], T[9])

CNOT | (U[0] ,U[6])
CNOT | (U[2] ,U[6])
CNOT | (U[3] ,U[6])
Toffoli | (T[1], U[6], S[0]) #z10
CNOT | (S[0], S[6])
CNOT | (S[0], S[7])
CNOT | (U[0] ,U[6])
CNOT | (U[2] ,U[6])
CNOT | (U[3] ,U[6])

CNOT | (T[0], T[9])
CNOT | (T[3], T[9])
CNOT | (T[1], T[9])
Toffoli | (T[9], U[2], S[0]) #z16
CNOT | (S[0], S[2])
CNOT | (T[0], T[9])
CNOT | (T[3], T[9])
CNOT | (T[1], T[9])

CNOT | (U[3], U[6])
CNOT | (T[0], T[9])
Toffoli | (T[9], U[6], S[0]) #z15
CNOT | (S[0], S[1])
CNOT | (S[0], S[2])
CNOT | (S[0], S[3])
CNOT | (S[0], S[4])
CNOT | (S[0], S[5])
CNOT | (S[0], S[6])
CNOT | (S[0], S[7])
CNOT | (U[3], U[6])
CNOT | (T[0], T[9])

CNOT | (U[2], U[6])
CNOT | (U[3], U[6])
Toffoli | (T[8], U[6], S[2]) #z17
CNOT | (U[3], U[6])
CNOT | (U[2], U[6])

Toffoli | (T[9], U[6], S[5]) #z11

```
    X | S[1]
    X | S[2]
    X | S[6]
    X | S[7]
Uncompute ( eng )
```

7 Appendix B

In this section, we show the detail of our quantum circuit with 22 qubits and 60 Toffoli gates. As pointed out in our **Observation 2**, we need at least 4 ancilla qubits to store the 4 values of $t_{29}, t_{33}, t_{37}, t_{40}$. However, we can not construct the quantum circuit for Sbox with only 4 ancilla qubits. In the following, we propose a quantum circuit with 22 qubits, including 6 ancilla qubits. Note that our second quantum circuit is similar to our first quantum circuit. The part of computing the values of s_0, $\cdots$, s_7 with the 4 values of $t_{29}, t_{33}, t_{37}, t_{40}$ in our second circuit is the same as our first circuit. As a result, we just show a description of our second quantum circuit with 6 ancilla qubits in the following pseudo code, where x_7, y_1, y_2, $\cdots$, y_{17} are the input to the non-linear function F and $T[0]$, $\cdots$, $T[5]$ are the 6 ancilla qubits. Note that we shall clean up the 6 ancilla qubits $T[0]$, $\cdots$, $T[5]$ in the end, which means we need $21 \times 2 + 18 = 60$ Toffoli gates in this circuit.

Algorithm 2 Output $t_{29}, t_{33}, t_{37}, t_{40}$ with 6 ancilla qubits

Require:

input, x_7, y_1, y_2, $\cdots$, y_{17};
input, $T[0]$, $\cdots$, $T[5]$;
for $0 \leq i \leq 5$ **do**
    $T[i] = 0$;
end for
$T[0] = Toffoli(y_{13}, y_{16}, T[0])$; # T[0]=t7
$T[0] = Toffoli(y_5, y_1, T[0])$; # T[0]=t9
$T[1] = Toffoli(y_9, y_{11}, T[1])$; # T[1]=t12
$T[1] = Toffoli(y_{14}, y_{17}, T[1])$; # T[1]=t14
$T[2] = CNOT(T[1], T[2])$; # T[2]=t14
$T[2] = CNOT(T[0], T[2])$; # T[2]=t19
$T[2] = CNOT(y_{19}, T[2])$; # T[2]=t23
$T[0] = Toffoli(y_5, y_1, T[0])$; # T[0]=t7
$T[0] = Toffoli(y_2, y_7, T[0])$; # T[0]=t11
$T[3] = Toffoli(y_{12}, y_{15}, T[3])$; # T[3]=t2
$T[3] = Toffoli(y_3, y_6, T[3])$; # T[3]=t4
$T[1] = CNOT(T[3], T[1])$; # T[1]=t17
$T[1] = CNOT(y_{20}, T[1])$; # T[1]=t21
$T[3] = Toffoli(y_3, y_6, T[3])$; # T[3]=t2
$T[3] = Toffoli(y_4, x_7, T[3])$; # T[3]=t6
$T[4] = Toffoli(y_9, y_{11}, T[4])$; # T[4]=t12
$T[4] = Toffoli(y_8, y_{10}, T[4])$; # T[4]=t16
$T[3] = CNOT(T[4], T[3])$; # T[3]=t18
$T[3] = CNOT(y_{19}, T[3])$; # T[3]=t22
$T[0] = CNOT(T[4], T[0])$; # T[0]=t20
$T[0] = CNOT(y_{18}, T[0])$; # T[0]=t24
$T[4] = Toffoli(y_8, y_{10}, T[4])$; # T[4]=t12
$T[4] = Toffoli(y_9, y_{11}, T[4])$; # T[4]=0
Here T[0]=t24, T[1]=t21, T[2]=t23, T[3]=t22, T[4]=0, T[5]=0.
$T[4] = Toffoli(T[2], T[1], T[4])$; # T[4]=t26
$T[2] = CNOT(T[0], T[2])$; # T[2]=t30
$T[4] = CNOT(T[3], T[4])$; # T[4]=t31
$T[5] = Toffoli(T[2], T[4], T[5])$; # T[5]=t32
$T[5] = CNOT(T[0], T[5])$; # T[5]=t33
$T[2] = CNOT(T[0], T[2])$; # T[2]=t23
$T[2] = CNOT(T[5], T[2])$; # T[2]=t34
$T[4] = CNOT(T[3], T[4])$; # T[4]=t26
$T[4] = CNOT(T[0], T[4])$; # T[4]=t27
$T[5] = CNOT(T[4], T[5])$; # T[5]=t35
$T[2] = Toffoli(T[0], T[5], T[2])$; # T[2]=t37
$T[1] = CNOT(T[3], T[1])$; # T[1]=t25
$T[3] = Toffoli(T[4], T[1], T[3])$; # T[3]=t29
$T[4] = Toffoli(T[0], T[5], T[4])$; # T[4]=t38
$T[1] = Toffoli(T[3], T[4], T[1])$; # T[1]=t40
$T[4] = Toffoli(T[0], T[5], T[4])$; # T[4]=t27
$T[5] = CNOT(T[4], T[5])$; # T[5]=t33