Journal of Artificial Intelligence and Autonomous Intelligence Received 20-10-2024; Accepted 21-12-2024; Published 28-12-2024

A Quine-McCluskey Based Method for Generating Optimum Combinational Logic Circuits from Reversible Quantum Circuits

Orhan Yaman orhanyaman@firat.edu.tr
Department of Digital Forensics Engineering
Firat University, Elazig, Turkey

Tuba Sanli 222144109@firat.edu.tr
Department of Digital Forensics Engineering
Firat University, Elazig, Turkey

Mehmet Karakose mkarakose@firat.edu.tr
Department of Computer Engineering
Firat University, Elazig, Turkey

Corresponding Author: Orhan Yaman

Abstract

Algorithms specifically designed for quantum computers have been developed. In quantum circuits, the Feynman, Toffoli, and Fredkin gates are employed instead of traditional inputs such as AND, OR, NAND, NOR, XOR, and XNOR in combinational logic gates. The ability to convert quantum circuits into combinational logic circuits, or vice versa, is of utmost importance. This essay study (or paper) aims to demonstrates the process of deriving combinational logic circuits from reversible quantum circuits. To achieve this, the Quine-McCluskey technique was utilized along with state tables generated from the quantum circuits to obtain an optimal logic expression that serves as the basis for constructing the combinational logic circuit. The resultant obtained combinational logic circuit was implemented within the MATLAB Simulink environment, and state tables were obtained. A comparison was made between the state tables derived from the quantum circuit and the combinational circuit, yielding successful results.

Keywords: Quantum circuits, Quine-McCluskey method, Quantum, Combinational logic Circuit, Optimization.

1. INTRODUCTION

Quantum computers are still in their early stages of development, but extensive research is being conducted to advance this technology. It is anticipated that in the future, with the progress of quantum technology, significantly faster quantum computers will emerge. These advanced quantum computers are believed to replace classical computers that are limited in their capabilities [1]. Quantum computing and reversible computing are inextricably linked, with the requirement that

Citation: Orhan Yaman, et al. A Quine-McCluskey Based Method for Generating Optimum Combinational Logic Circuits from Reversible Quantum Circuits. Journal of Artificial Intelligence and Autonomous Intelligence. Research. 2024;1(2):10. Pages: 139-154

all gates in quantum circuits must be reversible. Reversible gates possess the unique characteristic of preserving information without any loss, based on the input-output mapping. In contrast, traditional digital computer systems incur information loss and dissipate heat due to the erasure of vast amounts of information. This information loss and heat dissipation occur in all digital logic gates, including popular ones such as AND, NAND, NOR, EX-OR, and OR gates. Notably, the renowned scientist Bennettit was proposed that quantum circuits possess the remarkable ability to preserve information without any loss [2]. In the realm of quantum electronics theory, the fundamental unit of information is a qubit, whereas traditional computers employ bits, represented by '1' and '0'. This introductory section provides an overview of classical computer gates, quantum logic gates, the simulators utilized for gate synthesis, and a literature review to contextualize the information presented.

1.1 Combinational Logic Circuits and Gates

Combinational logic circuits, are a subcategory of digital electronic circuits, and are circuits that produce an instant output based on certain inputs. These circuits perform logical operations depending on the instantaneous states of the input signals. In combinational logic circuits, the output signal depends only on the current input signals and there is no memory or feedback in the circuit [3]. These circuits are circuits that perform basic logic operations and are built using a set of logic gates (AND, OR, NOT, etc.). Combinational logic circuits perform functions that produce a particular output according to the states of the inputs. For example, the state of one input may or may not contribute to an output signal depending on other inputs. Combinational logic circuits are used in many different applications. Computers, electronic calculators, data encoders and resolvers, multipliers, converters, and many more digital systems make use of combinational logic circuits. It is concerned with the design and analysis of combinatorial logic circuits, Boolean algebra, and logic circuit theory [3]. These circuits are often represented through symbolic logical diagrams or circuit drawings. The basic logic gates used in building combinational logic circuits are:

AND Gate: Receives multiple input signals and produces "1" as output if all inputs are "1". Otherwise, the output will be "0".

OR Gate: Receives multiple input signals and produces "1" as output if at least one input is "1". Output is "0" if all inputs are "0".

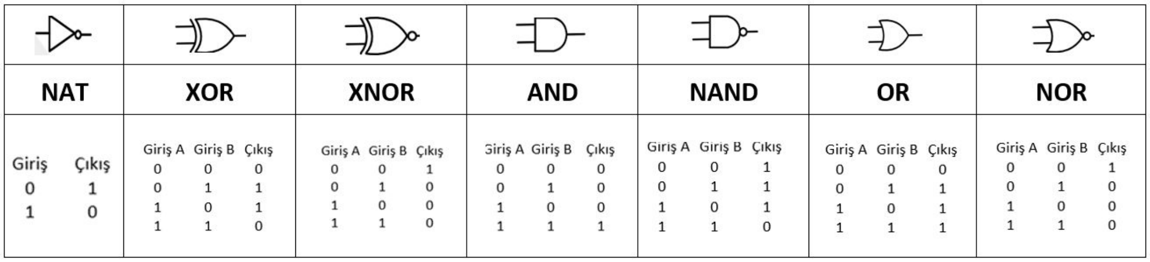
NOTE Gate: It takes a single input signal and produces the output that is complementary to the input. If the input is "1" the output will be "0", if the input is "0" the output will be "1".

XOR (Exclusive OR) Gate: It receives two input signals and produces "0" as output if the inputs are the same, and "1" if the inputs are different.

NAND (NOT AND) Gate: Receives multiple input signals and produces "0" as output if all inputs are "1". The output is "1" if at least one input is "0".

NOR (NOT OR) Gate: Receives multiple input signals and produces "0" as output if at least one input is "1". Output is "1" if all inputs are "0".

These basic logic gates are used in the design of combinational logic circuits to create circuits with different functions. In addition to, these basic gates can be used as well as more complex gates can be used as well. The classical standard symbols of the basic logic gates and their corresponding state tables are given shown in FIGURE 1. Truth tables show how a given logic gate works and produces output based on input signals. All possible combinations for input signals are listed and the corresponding output value is given for each combination. The XNOR gate is the complement of the XOR gate. The output is 1 when the inputs are the same, and 0 when the inputs are different. Therefore, the XNOR gate performs the “equality” function.

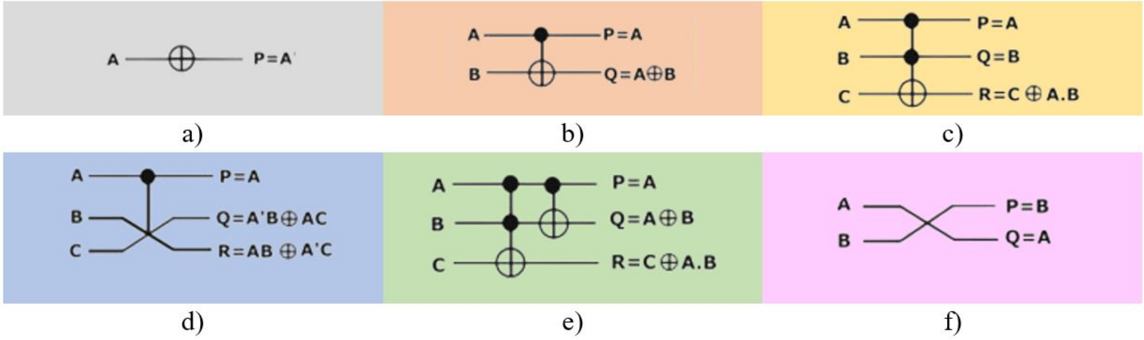


NAT	XOR	XNOR	AND	NAND	OR	NOR
Giriş Çıkış 0 1 1 0	Giriş A Giriş B Çıkış 0 0 0 0 1 1 1 0 1 1 1 0	Giriş A Giriş B Çıkış 0 0 1 0 1 0 1 0 0 1 1 0	Giriş A Giriş B Çıkış 0 0 0 0 1 0 1 0 0 1 1 1	Giriş A Giriş B Çıkış 0 0 1 0 1 1 1 0 1 1 1 0	Giriş A Giriş B Çıkış 0 0 0 0 1 1 1 0 1 1 1 1	Giriş A Giriş B Çıkış 0 0 1 0 1 0 1 0 0 1 1 0

Figure 1: Symbolic representation of example combinational logic gates *[2]*, and truth tables.

1.2 Quantum Logic Gates

Reversible gates, with the support of gate libraries developed over the years, can be exemplified include by the “Hadamard,” “Controlled V,” and “Controlled V+” [4], gates, as well as “NCT (NOT CNOT-TOFFOLI)” and “NCTSF (NOT-CNOT-TOFFOLI SWAP)-FREDKIN (controlled swap)” gates [5]. FIGURE 2 presents the representative symbols of these gates.



a) b) c)

d) e) f)

Figure 2: Reversible gates a) NOT b) CNOT c) TOFFOLI d) FREDKIN e) PERES f) SWAP [6].

1.3 Simulators Used in Programming Quantum Circuits

To design any reversible quantum circuit, programming, and design tools are required. Simulators used in quantum circuit programming are crucial tools for developing quantum computing algorithms, testing fault-tolerant quantum algorithms, and analyzing quantum circuit designs. Researchers and developers can conduct experiments using these simulators to better understand and optimize quantum computations, which can accelerate progress in the field of quantum information processing. Therefore, Thakral et al.[7], emphasized in the literature that the entire workflow of reversible logic and reversible logic circuit progresses throughout the synthesis, optimization, testing, and verification stages. They discussed and described frequently used simulators and software tools [7]. These simulators include RCViewer, REVKIT, RCTEST, RCDEV, QCADesigner, and RPGASim 2013. Pathak et al. [2], also mentioned additional simulators such as Qiskit (IBM), pyQuil (Google), and ProjectQ (ETH) in addition to other simulators [2].

Here are the functionalities of some commonly used quantum circuit simulators:

Qiskit [2]***:*** Qiskit is an open-source quantum programming simulator developed by IBM. It is a Python-based platform that provides access to the IBM Quantum Experience platform and supports connections to both various simulators and real quantum computers.

Cirq: Cirq, developed by Google, is a Python-based quantum programming library. Cirq enables the creation, simulation, and analysis of quantum circuits. Additionally, it provides direct access to Google's quantum computers.

Microsoft Quantum Development Kit: The Microsoft Quantum Development Kit is a simulation package that offers a range of tools and resources for quantum programming. This package allows the design and simulation of quantum circuits using the Q# programming language and facilitates the development of quantum algorithms.

QuTiP [2]***:*** QuTiP (Quantum Toolbox in Python) is an open-source quantum simulation package based on Python. QuTiP solves quantum mechanical equations and performs mathematical operations used for simulating quantum circuits.

ProjectQ [2]***:*** ProjectQ is an open-source quantum programming framework based on Python. It provides a high-level quantum programming interface and supports simulation as well as real quantum devices.

These simulators can be used for designing and analyzing quantum circuits, testing quantum algorithms, and simulating quantum computations. Additionally, most simulators offer access to real quantum computers, allowing the execution of algorithms on physical devices.

As shown in TABLE 1, the literature highlights some of the studies conducted over time and the simulators used.

Table 1: Uses of related simulators in the literature.

	RcViewer+	**QCViewer**	**Revkit**	**RCDEV**	**RCTEST**	**QCADesigner**	**Qiskit**
[8]	+	-	-	-	-	-	-
[9]	+	+	-	-	-	-	-
[10]	+	-	-	-	-	+	-
[2]	+	-	+	-	-	-	-
[11]	-	-	-	-	-	+	-
[12]	-	-	-	-	-	-	+
[13]	+	-	-	-	-	-	-

1.4 Literature Review

Before the connection between quantum computers and classical computers, foundational research in quantum computation, optimization, and deep learning-based artificial intelligence methods are presented. Nagamani et al. [8], designed a 4x4 Urdhva Tiryakbhayam Multiplier using Vedic mathematics techniques and Toffoli, Peres, and Fredkin gates. In their respective design, they used the RCViewer+ synthesis tool to compute the quantum cost, which resulted in 143 and 32 for the constant input, respectively [8]. Mısra et al. [10], aimed to reduce the quantum cost by utilizing the field of Quantum-dot Cellular Automata (QCA) in Nanoelectronics. They developed a new synthesis tool, QCADesigner, and proposed the use of DC gates. Consequently, they achieved a 25% reduction in quantum cost, and the gate count and garbage outputs were computed as 66% and 50%, respectively [10]. Sultana et al. [14], introduced methods such as Toffoli mapping, one-way algorithm, and Hamming distance-based complexity analysis using 16 4-variable gates. They also performed a quantum comparison among the gates used. The RPS gate exhibited the highest cost among the utilized gates [14]. Thakral et al. [7], implemented a quantum application using the Fredkin gate and subsequently provided a rapid guide for reversible logic gates in quantum studies. TABLE 2 presents further research conducted on the synthesis and optimization of quantum reversible circuits in the literature.

**T-depth: Depth of gate is a measure of how many ’layers’ of t-gate are executed in parallel, t-number: total number of t-gates in the quantum circuit, Qubit cost: Qubit Cost is Total Qubits Required to Design the Quantum. DFT: Design for Several Testability Using the Method of Converting a Standard Circuit to Its Testable Form, QCA: Quantum Dot Cellular Automaton, GC: Gate Count, QC: Quantum Cost, UD: Unit Delay, LC: Logical Computing, GO: Trash Out, CI: Fixed Input, QCADesigner: Design of Low-Cost Quantum Decoder in 1d Molecular-Qca, BE: Binary Redundancy, BG-2: Binary Gray, GB-2: Gray Binary, NG-R1: New Gate = New Gate, R = Reversible, TG: Two-Qubit Gate Number, R-CQCA: Reversible Conservative Quantum Cellular Automaton

In FIGURE 3, the quantum circuit equivalent of logical gates is given.

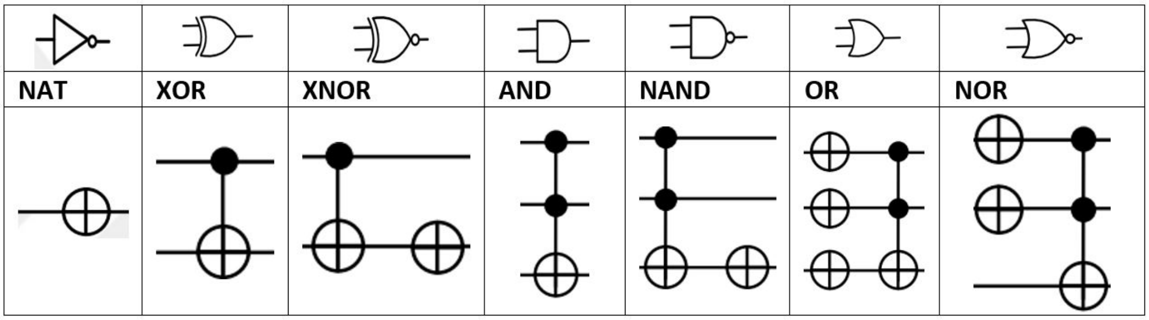


Figure 3: Quantum circuit equivalent of logic gates [2].

Table 2: Studies in the literature on the synthesis and optimization of quantum reversible circuits.

Scientific work	Year	Gate and Circuit Types	Recommended Method	Synthesis Tool	Results
[6]	2019	• Quantum circuits consisting of NOT, CNOT, and Toffoli gates (NCT library)	• Automatic Test Pattern Generation (ATPG) Method	-	• Addresses the problem of testing bridging faults in a reversible circuit. • 100% error rate
[15]	2019	• Toffoli gates	• A Synthesis of reversible Functions with Quantum cost Algorithm recommended	RCViewer+	QC: 7 GC: 3
[16]	2019	• EPOE expressions	• Recommended compact design of a general sequential circuit. • Designs of reversible sequential circuits.	RCViewer+	QC: 28 GO: 2 CL: 4
[17]	2020	• FG gate • HNG gate • BJN gate • PG gate • SCL etc.	• BCD collector application • Reversible implementation of the proposed design	RCViewer+	CI:14 GO:19 QC:63 GC:12 , Has a 10% improvement
[18]	2020	• Quantum equivalent structure • Controlled-V and +V gate structure	• KMD gates • Basic reversible gates	RCViewer+	KMD gate1 QC:9 KMD gate2 QC:10 KMD gate3 QC:9 KMD gate4 QC:22 MNFT QC:7 ZPLG QC:12
[19]	2020	• k-CNOT reversible circiut • Python 3.4 • LINDO Extended 17.00 software	• Full test set creation method • Complexity Analysis of the ILP Formulation	RCViewer+	SMGF ve MMGF error detections 100% error coverage
[11]	2021	• QCA • Logical crossover method	• Dual Port Memory(RAM)	QCADesigner	Decoder:%25.71 ACB:16.83 CLB:8.62 Data Router:4.74 Memory cell:3.73 CI:67, GO:50 QC:285, GC:57
[20]	2021	• CNOT, NOT gates • IBM'in QX architecture • Clifford+T gates	• Proposed design for detection and location in reversible sequential circuits. • LUT-based automatic error correction application	RCViewer+	QC: 14, GO:4 GC:4 SMGF, MMGF, PMGF, RGF, SBF, and SAF detected fault patterns.

1.5 Motivation

According to recent studies and research in the literature, there has been a significant emphasis focus on the optimization of reversible gates, specifically . These studies have focused on the development of quantum cost, garbage overhangs, and quantum circuits using various optical techniques, such as input and output optimizations, gate count reduction, and applying deep learning to circuits. As the studies progressed, researchers, also explored circuit faults and worked on enhancing their designs while creating quick guides for implementation. Furthermore, a comparative analysis was conducted to assess the disparities between quantum circuits and classical circuits, particularly in terms of their connection and combinational logic circuitry. However, it is worth noting that there is a limited number of studies focusing on the shared characteristics and relevant structures between quantum circuits and classical circuits. Despite the significant attention given to optimizing reversible gates and fault detection, there remains a gap in the research concerning their common aspects and interrelationships. Further exploration in this area could lead to valuable insights and advancements in the field of quantum computing. The motivation behind conducting these analyzes stemmed from a realization of the potential benefits and advancements that could be achieved through a comprehensive understanding of the common features and interconnections between quantum circuits and classical circuits. By identifying and exploring such shared characteristics, researchers sought to unlock new opportunities for improving both quantum and classical circuit designs. This pursuit of knowledge aimed to pave the way for more efficient and effective quantum computing methodologies, ultimately driving progress in the field.

2. MATERIAL

In the research, the RCViwer+ simulator, Quine-McCluskey monitoring, and MATLAB Simulink program [21], were utilized as tools. RCViewer+ was chosen due to its beneficial synthesis features and the ability to obtain images of the quantum circuit within the program.

2.1 Rcviewer+ Synthesis Tool

The RCViewer+ synthesis tool is a popular simulator tool using many quantum processors made known in the literature. Indeed, one of these studies is Thakral et al. [13], They stated that the RCViewer+ synthesis tool is a simple application software written in C++.TFC format [13]. The simulator has important capabilities such as calculating quantum cost and designing a quantum circuit consisting of reversible logic-based gates with optimized inputs, outputs, and gates. As a simulation tool for the development and analysis of quantum computing applications, this simulator constitutes an important tool in evaluating the performance of complex quantum algorithms and circuits. The simulation results can help researchers predict the resources and time required for quantum computations. During the development of quantum applications, the relevant programs should be written in the notebook and saved in “tfc” format, so that the simulator can be used effectively in the analysis and optimization of these applications.”

2.2 Quine-McCluskey Algorithm

The Quine-McCluskey algorithm can minimize logical associations for any number of inputs [22]. It starts with the truth table and ensures that all “product terms” (all combinations of inputs that produce a real output) are extracted. After provisioning it groups all “product terms” by the number of “ones” they contain. It then combines the “product terms” from adjacent groups. Sarkar et al. [23], incorporated “ant colony optimization (ACO)” and “simulated” annealing (SA)” into the Quine-McCluskey approach [23]. In the proposed method, obtaining logical expressions using the classical Boolean logic synthesis method has helped to design classical circuits (combinational logic circuits). FIGURE 6 and the following describe the steps of the proposed algorithm.

3. PROPOSED METHOD

In this study, the Our proposed method consists of three basic steps: (i) the . These steps consist of the quantum step, (ii) the Quine-McCluskey Algorithm, and (iii) the classical (combinational logic circuits) steps. The general block diagram of the proposed method is given in FIGURE 4.

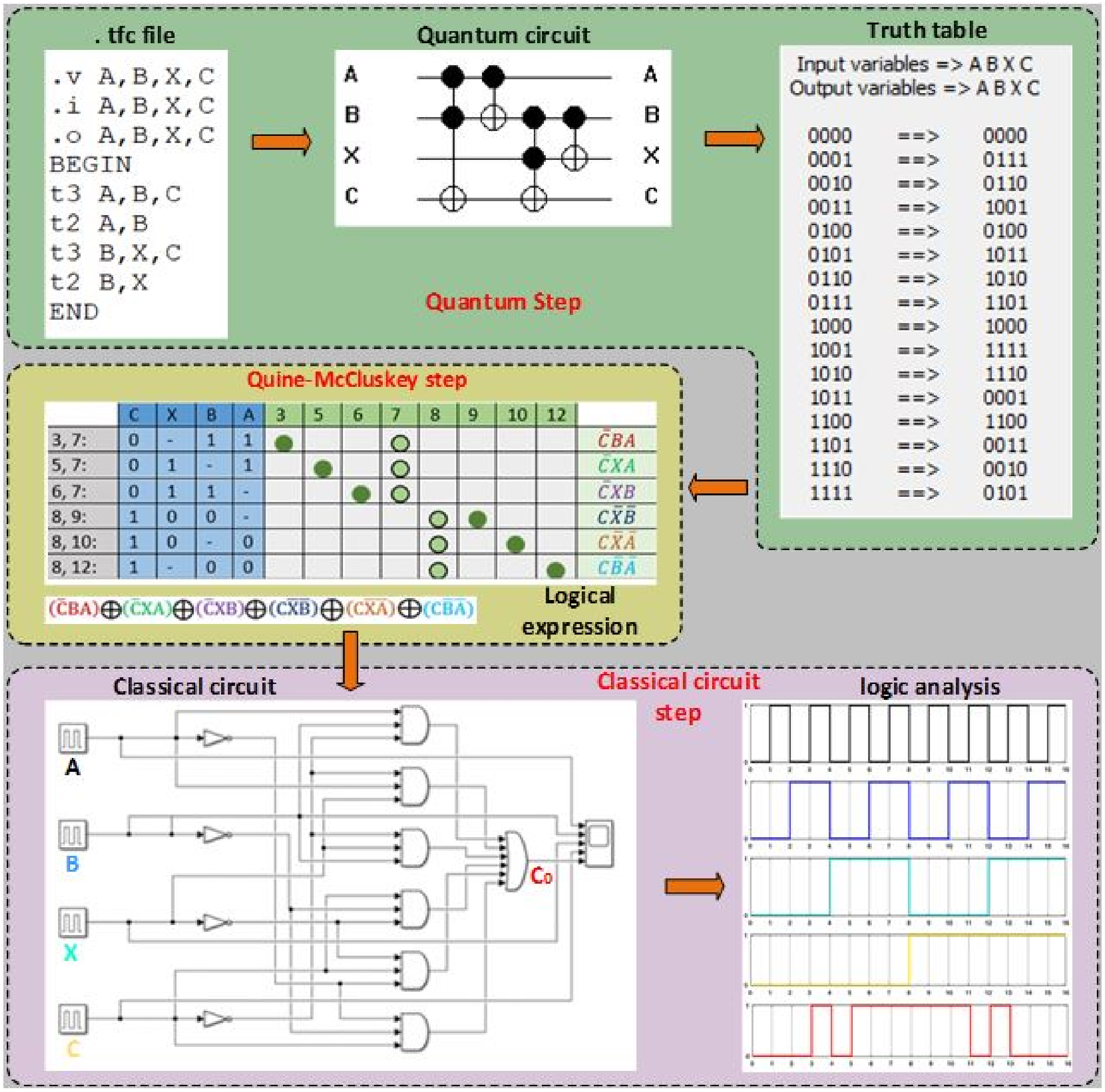
3.1 Constructing the State Table of a Quantum Circuit

In this step, a sample “.tfc” file was created with the RCViewer+ simulator installed on the classical computer, and then a 4-input-output and 4-port quantum circuit was designed. As seen in FIGURE 5, shows, the truth table of the obtained circuit was taken. Simulation tools or quantum computers can be used to generate quantum truth tables. Simulation tools can obtain truth tables by calculating the outputs of quantum circuits. In the study, the RCViewer+ simulator was used for quantum circuit design. The reason why this simulator is preferred is that the generated “.tfc” files can be converted into quantum circuits. Real quantum computers, on the other hand, run quantum circuits in real-time to obtain the outputs. Quantum truth tables are an important tool for the analysis and design of quantum computations. These tables help users analyze their circuits to ensure their quantum circuits work correctly and achieve the desired results.

3.2 Obtaining Logical Expression with Quine-McCluskey Algorithm

In this part of the application, the states where the value of output ’C’ is equal to 1 are determined by taking the state table of a circuit. Then, this output value is replaced with ’Y’ and a logical expression is obtained using the Quine-McCluskey algorithm. The resulting logical expression is then converted to its quantum logical equivalent. FIGURE 6 contains the steps showing how to obtain the equivalent logical expression using the Quine-McCluskey algorithm from the state table of a quantum circuit.

Basic prime effects extracted from classical gates obtained for the example quantum circuit given in FIGURE 6; $(\overline{C}BA)$, $(\overline{C}XA)$, $(\overline{C}XB)$, $(C\overline{X}\overline{B})$, $(C\overline{X}\overline{A})$, $(C\overline{B}\overline{A})$. Minimum boolean equation and



```
.v A,B,X,C
.i A,B,X,C
.o A,B,X,C
BEGIN
t3 A,B,C
t2 A,B
t3 B,X,C
t2 B,X
END
```

```
Input variables => A B X C
Output variables => A B X C

0000    ==>    0000
0001    ==>    0111
0010    ==>    0110
0011    ==>    1001
0100    ==>    0100
0101    ==>    1011
0110    ==>    1010
0111    ==>    1101
1000    ==>    1000
1001    ==>    1111
1010    ==>    1110
1011    ==>    0001
1100    ==>    1100
1101    ==>    0011
1110    ==>    0010
1111    ==>    0101
```

	C	X	B	A	3	5	6	7	8	9	10	12	
3, 7:	0	-	1	1	●			○					$\bar{C}BA$
5, 7:	0	1	-	1		●		○					$\bar{C}XA$
6, 7:	0	1	1	-			●	○					$\bar{C}XB$
8, 9:	1	0	0	-					○	●			$C\bar{X}\bar{B}$
8, 10:	1	0	-	0					○		●		$C\bar{X}\bar{A}$
8, 12:	1	-	0	0					○			●	$C\bar{B}\bar{A}$

$(\bar{C}BA)\oplus(\bar{C}XA)\oplus(\bar{C}XB)\oplus(C\overline{XB})\oplus(C\overline{XA})\oplus(C\overline{BA})$

Figure 4: General block diagram of the proposed method.

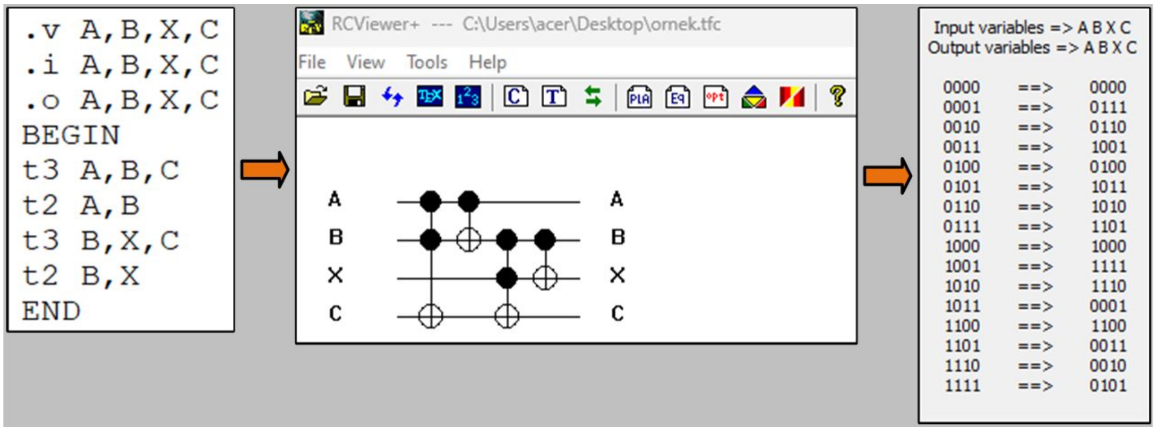
its conversion to the optimum logical equivalent expression:

$$y = (\overline{C}BA)\ V(\overline{C}XA)\ V(\overline{C}XB)\ V(C\overline{XB})\ V(C\overline{XA})V\ (C\overline{BA}) \tag{1}$$

$(\overline{C}BA)\oplus(\overline{C}XA)\oplus(\overline{C}XB)\oplus(C\overline{XB})\oplus(C\overline{XA})V\ (C\overline{BA})$ is in the form.

3.3 Generating Combinational Logic Circuit from Logical Expression

In this last part of the study, a combinational logic circuit is designed using the converted equivalent expression and the MATLAB Simülink program installed on the classical computer. The logic analysis of the designed circuit is presented in FIGURE 7.

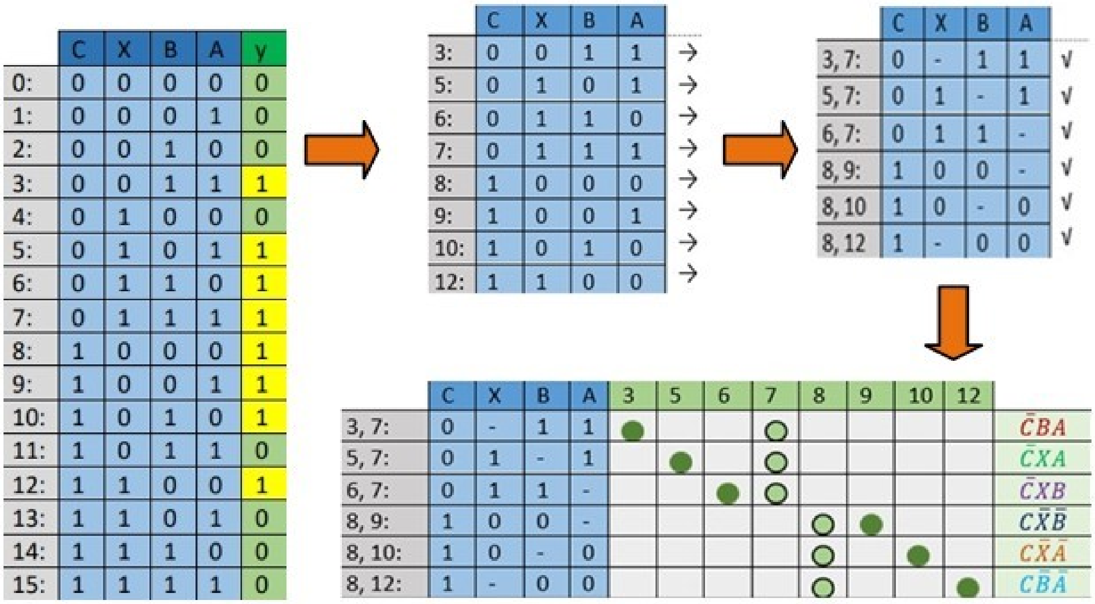


```
.v A,B,X,C
.i A,B,X,C
.o A,B,X,C
BEGIN
t3 A,B,C
t2 A,B
t3 B,X,C
t2 B,X
END
```

```
Input variables => A B X C
Output variables => A B X C

0000    ==>    0000
0001    ==>    0111
0010    ==>    0110
0011    ==>    1001
0100    ==>    0100
0101    ==>    1011
0110    ==>    1010
0111    ==>    1101
1000    ==>    1000
1001    ==>    1111
1010    ==>    1110
1011    ==>    0001
1100    ==>    1100
1101    ==>    0011
1110    ==>    0010
1111    ==>    0101
```

Figure 5: The result of the quantum circuit design and truth table of the sample ".tfc" file.



	C	X	B	A	y
0:	0	0	0	0	0
1:	0	0	0	1	0
2:	0	0	1	0	0
3:	0	0	1	1	1
4:	0	1	0	0	0
5:	0	1	0	1	1
6:	0	1	1	0	1
7:	0	1	1	1	1
8:	1	0	0	0	1
9:	1	0	0	1	1
10:	1	0	1	0	1
11:	1	0	1	1	0
12:	1	1	0	0	1
13:	1	1	0	1	0
14:	1	1	1	0	0
15:	1	1	1	1	0

	C	X	B	A	
3:	0	0	1	1	→
5:	0	1	0	1	→
6:	0	1	1	0	→
7:	0	1	1	1	→
8:	1	0	0	0	→
9:	1	0	0	1	→
10:	1	0	1	0	→
12:	1	1	0	0	→

	C	X	B	A	
3, 7:	0	-	1	1	√
5, 7:	0	1	-	1	√
6, 7:	0	1	1	-	√
8, 9:	1	0	0	-	√
8, 10	1	0	-	0	√
8, 12	1	-	0	0	√

	C	X	B	A	3	5	6	7	8	9	10	12	
3, 7:	0	-	1	1	●			○					$\bar{C}BA$
5, 7:	0	1	-	1		●		○					$\bar{C}XA$
6, 7:	0	1	1	-			●	○					$\bar{C}XB$
8, 9:	1	0	0	-					○	●			$C\bar{X}\bar{B}$
8, 10:	1	0	-	0					○		●		$C\bar{X}\bar{A}$
8, 12:	1	-	0	0					○			●	$C\bar{B}\bar{A}$

Figure 6: Conversion of sample state table to logical equivalent expression using Quine-McCluskey.

FIGURE 7 shows the results analysis of the combinatorial circuit obtained with the MATLAB Simulink program. 'C0' outputs produced for 'A', 'B', 'X', and 'C' inputs are given. The results obtained with the combinatorial circuit are compared with the results obtained with the equivalent quantum circuit.

4. EXPERIMENTAL RESULTS

This study examines the quantum truth table showing the effect of quantum gates and quantum circuit operations applied to the input bits of a quantum circuit on the output bits. Unlike classical

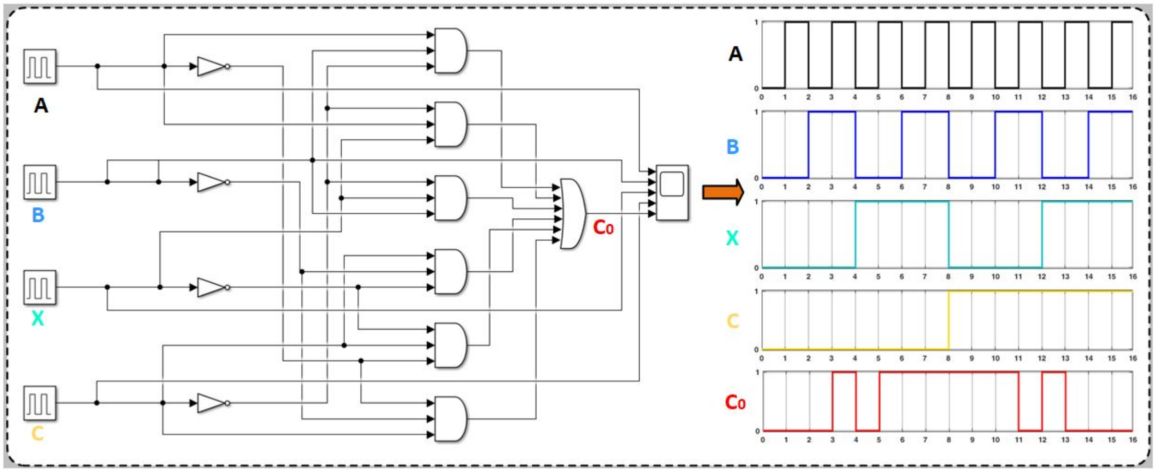


Figure 7: Classical circuit and logic analysis produced.

truth tables, these quantum truth tables also include the effect of properties such as quantum superposition and quantum parallelism. The quantum truth table presents the probability distributions of the output bits of a quantum circuit, , and the probability distributions of the output bits are calculated for each input combination. This table is used to analyze a quantum circuit's behavior and predict the results. TABLE 3 presents the effect of the truth table obtained as a result offrom the quantum circuit design of the study on the output bits and the details of these values. The use of such truth tables for analysis and performance evaluations of quantum circuits is important to provide a better understanding of the field of quantum computing and contribute to optimizing quantum circuits. This study can be considered a fundamental step in the development of quantum computing technologies and designing quantum-computing algorithms. Moreover, the results obtained can guide the development of future quantum circuit designs and support progress in the field of quantum computing.

There are fundamental differences between classical circuits (combinational logic circuits) and quantum circuits. Classical circuits use classical bits, while quantum circuits use qubits and represent information using quantum superposition and quantum interactions. In quantum circuits, operations are performed with quantum gates, and results are obtained by measuring qubits. These differences reflect the fundamental differences between classical and quantum computing.

There are some common features between classical circuits and quantum circuits, including:. Here are some of these features:

- Logical Operations: Both types of circuits can perform logical operations. Logical operations produce outputs with certain operations applied to the inputs. These operations are performed with logical gates or circuit elements in classical circuits and quantum gates in quantum circuits.
- Inputs and Outputs: Both types of circuits work with inputs and outputs. Inputs are signals supplied from outside the circuit and outputs represent the result of the circuit.

Table 3: Quantum truth table.

Input				**Output**			
C	**X**	**B**	**A**	$\mathbf{C}_0$	$\mathbf{X}_0$	$\mathbf{B}_0$	$\mathbf{A}_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	1	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	1	0	1

- *Design Methodology*: Both classical circuits and quantum circuits are based on a specific design methodology. These methodologies include steps such as the selection of circuit elements, the configuration of the circuit, and the sequencing of operations.
- *Error Analysis and Error Correction*: Both classical circuits and quantum circuits are subject to error analysis and error correction techniques. Detecting and correcting errors increases the reliability of the circuit.
- *Optimization and Efficiency*: Both types of circuits can be evaluated in terms of optimization and efficiency. Factors such as the size and complexity of the circuits and the efficient use of resources affect the optimization of the design.
- *Simulation and Analysis*: The performance and functionality of classical circuits and quantum circuits can be analyzed with simulation tools. These analyzes are important to ensure that the circuit works as expected.

In this study, the common features between classical and quantum circuits based on the basic principles of electronic circuits are emphasized. Both types of circuits are used in the design, analysis, and optimization of electronic systems. However, quantum circuits are used in specialized fields, especially in applications such as quantum computing and quantum computing. In addition-Additionally, to these differences and common features, the connection of the circuits with each other was determined by the method proposed in the study and illustrated in (FIGURE 8).

The proposed method of the study includes a research and analysis approach used to understand the interconnections of between classical and quantum circuits. FIGURE 8 provides a presentation

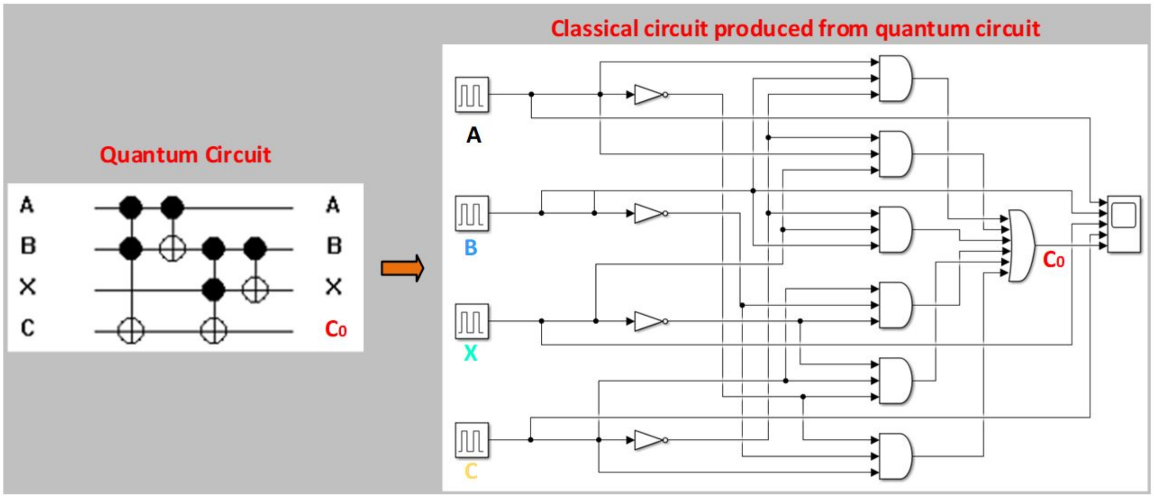


Figure 8: The classical circuit produced from the designed quantum circuit.

that visualizesillustrates these interconnections, explaining how circuits can interact with each other and how quantum circuits can be related to classical circuits. Such research can contribute to the understanding of the interaction between quantum and classical systems, as well as advance in the field of quantum computing. In conclusion, this study can be considered as an important step to understand and improve the use of electronic circuits in various fields.

5. DISCUSSION

A quantum truth table is a tool that visualizes the experimental results of quantum circuits. This table (TABLE 4) presents the effect of quantum gates applied to the input bits of the quantum circuit on the output bits and the probability distributions of these bits. This is used to analyze the behavior of the quantum circuit and predict its consequences.

Comparison of experimental results is important to determine the advantages or disadvantages of quantum circuits over classical circuits. Comparing the performance of quantum circuits with classical circuits, especially for quantum computing problems, is critical to understand what advantages they can provide in potential applications.

6. CONCLUSIONS

Experimental results show the accuracy of the proposed Quine-McCluskey algorithm-based method for transforming the quantum circuits shown in FIGURE 8, into classical reversible circuits. In addition, the comparison results in TABLE 4, reveal that it is possible to construct classical circuits from the truth table of quantum circuits of the results obtained with the proposed method. The method proposed in this study aims to provides a possiblegive direction to future studies by discussing in general terms and aims to contribute to the development and direction of quantum theory to quantum

Table 4: Comparison of experimental results of the classical circuit and quantum circuit.

Input Signals	Combinational logic circuit result signals	Quantum circuit result signals
0000 ==>	0	0
0001 ==>	0	0
0010 ==>	0	0
0011 ==>	1	1
0100 ==>	0	0
0101 ==>	1	1
0110 ==>	1	1
0111 ==>	1	1
1001 ==>	1	1
1010 ==>	1	1
1011 ==>	0	0
1100 ==>	1	1
1101 ==>	0	0
1110 ==>	0	0
1111 ==>	0	0

computers. This study aims toThe study proves the efficiency and accuracy of a method used to associate quantum circuits with classical circuits. In the field of quantum computing, it is important to develop and analyze quantum circuits by taking advantage of classical circuits. In particular, the use of classical methods such as the Quine-McCluskey algorithm in the optimization and design of quantum circuits may contribute to the advancement of quantum computing technologies. In future studies, performing more detailed analyzes on this proposed method and applying it to different quantum circuits may help us better understand the general validity and effectiveness of the method. Further, a deeper examination of the relationship between classical and quantum circuits will help us better understand the potential of quantum theory beyond classical computers and strengthen the contributions of quantum computing technologies for future applications.

7. ACKNOWLEDGEMENTS

This study was supported by TUBITAK. Project number: 121E439.

References

[1] Yetiş H, Karaköse M. Variational Quantum Circuits for Convolution and Window-Based Image Processing Applications. Quantum Science and Technology (QST). 2023;8:045004.

[2] Pathak N, Misra NK, Bhoi BK, Kumar S. Concept and Algorithm of Quantum Computing During Pandemic Situation of COVID-19. In Smart Systems: Innovations in Computing: Proceedings of SSIC. Springer Singapore. 2022;235:523-535.

[3] Yetiş H, Karaköse M. Quantum Circuits for Binary Convolution. In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI). IEEE. 2020.

[4] Yetiş H, Karaköse M. An Improved and Cost Reduced Quantum Circuit Generator Approach for Image Encoding Applications. Quantum Inf. Process. 2022 ;21:203.

[5] Bar NF, Yetis H, Karakose M. An Efficient and Scalable Variational Quantum Circuits Approach for Deep Reinforcement Learning. Quantum Inf. Process. 2023;22:300.

[6] Handique M, Biswas S, Deka JK. Test Generation for Bridging Faults in Reversible Circuits Using Path-Level Expressions. Electron. Test. 2019;35:441-457.

[7] Thakral S, Bansal D. A Quick Guide to Implement Reversible Logic. In 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE. 2018.

[8] Nagamani AN, Prasad HV, Hathwar RS, Agrawal VK. Design of Optimized Reversible Multiplier for High Speed DSP Application. In 2015 10th International Conference on Information, Communications and Signal Processing (ICICS). IEEE. 2015.

[9] Gaur HM, Singh AK, Ghanekar U. A New DFT Methodology for k-CNOT Reversible Circuits and Its Implementation Using Quantum-Dot Cellular Automata. Optik. 2016;127:10593-10601.

[10] Misra NK, Sen B, Wairya S, Bhoi B. Testable Novel Parity-Preserving Reversible Gate and Low-Cost Quantum Decoder Design in 1D Molecular-QCA. J. Circuits Syst. Comput. 2017;26:1750145.

[11] Arunachalam K, Perumalsamy M, Ramasamy A. Multi-Port Memory Design in Quantum Cellular Automata Using Logical Crossing. J. Microelectron. Int. 2021;51:49-61.

[12] Yuan S, Gao S, Wen C, Wang Y, Qu H, et al. A Novel Fault-Tolerant Quantum Divider and Its Simulation. Quantum Inf. Process. 2022;21:182.

[13] Thakral S, Manhas P, Verma J. Quantum Implementation of Reversible Logic Gates Using RCViewer + Tool. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS. Springer Nature Singapore. 2022;1:409-418.

[14] Sultana M, Prasad M, Roy P, Sarkar S, Das S, et al. Comprehensive Quantum Analysis of Existing Four Variable Reversible Gates. In 2017 Devices for Integrated Circuit (DevIC). IEEE. 2017.

[15] Thakral S, Bansal D. Optimized Quantum Implementation Approach. In 2019 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA). IEEE. 2019.

[16] Kalantari Z, Eshghi M, Mohammadi M, Jassbi S. Low-Cost and Compact Design Method for Reversible Sequential Circuits. J. Supercomput. 2019 ;75:7497-7519.

[17] Thabah SD, Saha P. Low Quantum Cost Realization of Reversible Binary-Coded-Decimal Adder. Procedia Comput. Sci. 2020;167:1437-1443.

[18] Kamaraj A, Marichamy P, Kaviyashri KP. Realization and Optimization of Quantum Equivalent Circuits of Reversible Combinational Circuits. J. Comput. Theor. Nanosci. 2020;17:2080-2084.

[19] Handique M, Deka JK, Biswas S. An Efficient Test Set Construction Scheme for Multiple Missing-Gate Faults in Reversible Circuits. Electron. Test. 2020;36:105-122.

[20] Kheirandish D, Haghparast M, Reshadi M, Hosseinzadeh M. Efficient Techniques for Fault Detection and Location of Multiple Controlled Toffoli-Based Reversible Circuit. Quantum Inf. Process. 2021;20:370.

[21] Yetiş H, Karaköse M. A New Framework Containing Convolution and Pooling Circuits for Image Processing and Deep Learning Applications With Quantum Computing Implementation. Traitement du Signal (TS). 2022;39:501-512.

[22] Yetis H, Karaköse M. A New Framework for Quantum Image Processing and Application of Binary Template Matching. In 2022 26th International Conference on Information Technology (IT). IEEE. 2022.

[23] Sarkar M, Ghosal P, Mohanty SP. Reversible Circuit Synthesis Using ACO and SA Based Quine-McCluskey Method. In 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE.2013.