frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 22 April 2025
DOI 10.3389/fphy.2025.1582819

OPEN ACCESS

EDITED BY
Nanrun Zhou,
Shanghai University of Engineering Sciences, China

REVIEWED BY
Lihua Gong,
Shanghai University of Engineering Sciences, China
Yefeng He,
Xi'an University of Post and Telecommunications, China

*CORRESPONDENCE
Fei Gao,
gaof@bupt.edu.cn

RECEIVED 25 February 2025
ACCEPTED 07 March 2025
PUBLISHED 22 April 2025

CITATION
Jiang L-L, Cai B-B, Gao F, Qin S-J, Jin Z-P and Wen Q-Y (2025) Constructing resource-efficient quantum circuits for AES. *Front. Phys.* 13:1582819. doi: 10.3389/fphy.2025.1582819

Constructing resource-efficient quantum circuits for AES

Liao-Liang Jiang[1,2], Bin-Bin Cai[3], Fei Gao[1,2,4]*, Su-Juan Qin[1,2,4], Zheng-Ping Jin[1,2,4] and Qiao-Yan Wen[1,2,4]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China, [2]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China, [3]College of Computer and Cyber Security, Fujian Normal University, Fuzhou, China, [4]National Engineering Research Center of Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing, China

An efficient quantum implementation of the advanced encryption standard (AES) is crucial for reducing the complexity of implementing an exhaustive key search through Grover's algorithm. In this paper, we study how to construct resource-efficient quantum circuits for AES. We consider the product of T-gates depth and width (TDW) and the product of full depth and width (FDW) as optimization targets. We propose a generic method, called the controlled control qubit cascade (CCQC) technique, to construct quantum circuits for nonlinear components with reduced TDW and FDW. Using this, we construct a quantum circuit for the AES S-box. Compared with recent work presented at ASIACRYPT 2023, our S-box quantum circuit achieves reductions of 2.3% in TDW and 45.2% in FDW. Additionally, we propose a new key schedule strategy to reduce the full depth of the AES quantum circuit. Finally, the trade-offs between T-gates depth and width and the parallel numbers of S-box and TDW are analyzed.

KEYWORDS

Grover algorithm, AES, S-box, quantum circuit, quantum resource optimization

1 Introduction

Quantum technology, including cryptography Shor and Preskill [1]; Qin et al. [2], quantum computing Grover [3]; Shor [4], and quantum precision measurement Braginsky and Khalili [5]; Childs et al. [6], is the frontier of a new round of scientific and technological revolution. Quantum computing has been widely applied in fields including quantum cryptography Zhou et al. [7]; Gong et al. [8], quantum simulation Buluta and Nori [9], machine learning Song et al. [10, 11]; Li et al. [12, 13], solving equations Childs et al. [14]; Wan et al. [15, 16], and cryptanalysis Kaplan et al. [17]; Cai et al. [18]. The threats that quantum computing poses to existing cryptographic systems are well-known. Once large-scale quantum computers become operational, Shor's algorithm will be capable of breaking asymmetric cryptographic schemes that rely on the discrete logarithm and factoring problems, such as RSA, ECDH, and ECC Shor [4]; Shor [19]. For symmetric cryptography, the primary challenges posed by quantum computation arise from Simon's algorithm and Grover's algorithm. When a quantum oracle that implements the target cryptographic quantum circuit can be

accessed, Simon's algorithm can undermine various symmetric cryptographic schemes by finding hidden periods Kaplan et al. [17]; Bonnetain et al. [20], while Grover's algorithm can provide quadratic acceleration for exhaustive key searches that attack block ciphers Grover [3].

Due to the extremely high cost of quantum resources, estimating the quantum resources needed to attack block ciphers using Grover's algorithm is crucial for reducing the implementation difficulty of such attacks and accurately predicting their actual implementation time. The advanced encryption standard (AES) Daemen [21] is one of the most widely used block ciphers today; thus, evaluating the quantum resources needed to attack it is highly significant. Moreover, the National Institute of Standards and Technology (NIST) has proposed the standardization of post-quantum cryptography by defining security categories 1, 3, and 5 based on the computational resources needed for exhaustive key searches on AES-128, AES-192, and AES-256, respectively. The key to applying Grover's algorithm to AES lies in implementing the Grover oracle, which utilizes the AES quantum circuit to mark the correct key during the search process. Consequently, the precise estimation and careful optimization of the quantum circuit implementing AES have attracted significant attention in recent years.

Computational resources are often measured by quantum circuit size. Metrics commonly used to measure quantum circuits include width, depth, and the number of quantum gates Specifically, width refers to the number of logical qubits needed in a quantum circuit. Meanwhile, the minimum stages of quantum gates that can be executed in parallel in a circuit is called depth. We can measure the depth of all elementary gates within the circuit or focus specifically on the depth of a particular quantum gate, depending on the requirements of our demand. From a physical implementation perspective, realizing quantum circuits with a large width or deep depth is quite difficult Sun et al. [22]. Therefore, prior works have often focused on ways to reduce either the width Grassl et al. [23]; Zou et al. [24]; Li et al. [25] or depth Jaques et al. [26]; Huang and Sun [27] required for Grover's attack on AES. However, there is often a trade-off between these two metrics. For example, optimizing the width of a quantum circuit may lead to a very large depth, which in turn makes Grover's attack difficult to implement. Hence, it is also very feasible to consider the product of width and depth as a metric for measuring the size of a quantum circuit. Specifically, because the running time of fault-tolerant quantum computers is proportional to the T-gates depth Fowler [28]; Amy et al. [29, 30], the T-gates depth is also commonly used as an optimization target. For distinction, this paper refers to the depth of all elementary gates as the F-depth and to the T-gates depth as the T-depth. Considering the significance of F-depth and T-depth, we introduce the definitions of FDW as the product of F-depth and width and TDW as the product of T-depth and width, respectively.

1.1 Related works

One of the key challenges in implementing the AES quantum circuit is constructing the quantum circuit for the nonlinear component S-box. Previous works can generally be categorized into three types based on optimization goals for S-box quantum circuits: reducing width, depth, or the product of width and depth. We introduce related works from these three perspectives.

1.1.1 Reducing width

This field was initiated by Grassl et al., who utilized the Itoh–Tsujii algorithm to find the multiplicative inverse in a finite field $GF(2^8)$ for an S-box Grassl et al. [23]. Consequently, extensive subsequent works Chung et al. [31]; Wang et al. [32]; Li et al. [25] focused on efficiently solving the multiplicative inverse in $GF(2^8)$. These works leveraged the properties of tower fields to map elements from an extension field to a subfield, thereby leading to the design of S-box circuits with low widths. Alternatively, Zou et al. [24] and Huang and Sun [27] employed the classical S-box circuit proposed by Boyar and Peralta [33] to construct S-box quantum circuits with low width, leveraging the fact that this classical S-box circuit's multiplicative complexity was optimized by a heuristic algorithm. Huang and Sun [27] developed an in-place circuit structure for an S-box, and Li et al. [34] utilized this circuit to construct an S-box quantum circuit with five ancilla qubits.

1.1.2 Reducing depth

Jang et al. [35] constructed the first S-box quantum circuit aimed at reducing depth. However, the focus was usually on T-depth. Huang and Sun [27] formulated a technique that transforms a classical circuit with a multiplicative depth of t into a quantum circuit with a T-depth of t. They utilized the S-box classical circuit Boyar and Peralta [36] to construct an S-box quantum circuit aimed at optimizing T-depth. Furthermore, Huang et al. optimized the classical S-box circuit Boyar and Peralta [36] and constructed an S-box quantum circuit with the theoretically lowest T-depth of 3 using this optimized classical circuit. Additionally, Huang et al. provided theoretical proof that the T-depth equals 3.

1.1.3 Reducing the product of width and depth

This metric was first presented and optimized by Jaques et al. Jang et al. [35], but fewer studies focused on it. Most recently, the product of width and depth has begun to receive attention. Liu et al. proposed a technique called m-XOR at ASIACRYPT 2023 Liu et al. [37] that can identify reusable qubits. They also designed a compact circuit structure for the S-box and constructed an S-box quantum circuit with a product of T-depth and width equal to 344.

1.2 Our contributions

We analyze the algebraic structures of S-box classical circuits, uncovering the intrinsic connections between multiplicative nodes. This inspires our controlled control qubit cascade (CCQC) technique for constructing quantum circuits with optimized TDW and FDW. Applying CCQC, our S-box quantum circuit reduces TDW by 2.3% and FDW by 45.2% compared to ASIACRYPT 2023 work Liu et al. [37]. We develop a key schedule strategy to reduce AES circuit F-depth. With this and our S-box circuit, we estimate the quantum resources required to implement iterative AES-128. Finally, we analyze the trade-offs of T-depth and width, as well as S-box parallelism versus TDW. Finally, we analyze T-depth vs. width and S-box parallelism vs. TDW trade-offs.

2 Preliminaries

2.1 Synthesis of quantum circuits

A qubit is typically denoted as $|q\rangle$, and $|q\rangle^{\otimes n}$ refers to a n-qubit quantum system that can also be represented by unit vectors in $\mathbb{C}^{2^n}$ Nielsen and Chuang [38]. A quantum circuit transforms the initial input state into the final output state through a series of unitary operations, where the unitary transformation U is a linear map and satisfies $U \cdot U^{\dagger} = I$, and $U^{\dagger}$ is the adjoint of U. Any unitary transformation can be constructed by a composition and a tensor product of a universal gate set. A universal gate set consists of a finite number of single-qubit gates and two-qubit gates. And $U^{\dagger}$ can be obtained by reversing the order of adjoint gates in U. Quantum circuits require that ancilla qubits should be ultimately returned to $|0\rangle$; thus, $U^{\dagger}$ is typically used for uncompute operations. We adopt the Clifford + T set as this gate set that can be implemented fault-tolerantly on a large set of surface codes. The Clifford + T set includes

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{i} \end{pmatrix}, \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

and a non-Clifford gate $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\mathbf{i}\pi/4} \end{pmatrix}$, where $\mathbf{i} = \sqrt{-1}$. We also apply the Pauli-X gate $X = HS^2H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, which can implement $|a\rangle \rightarrow |a+1\rangle$. Here, "+" implies XOR. The CNOT gate implements $|a\rangle|b\rangle \rightarrow |a\rangle|a+b\rangle$, and the Toffoli gate implements $|a\rangle|b\rangle|c\rangle \rightarrow |a\rangle|b\rangle|c+a\cdot b\rangle$. In quantum circuits, X gates, CNOT gates, and Toffoli gates, respectively, perform the corresponding classical NOT, XOR, and operations. The quantum AND gate is similar with the Toffoli gate to $|a\rangle|b\rangle|0\rangle \rightarrow |a\rangle|b\rangle|a\cdot b\rangle$. The difference is that the quantum AND gate requires the input state of the target qubit as $|0\rangle$. In this paper, the quantum circuit we constructed perfectly meets this condition. Hence, we adopt a quantum AND gate because the T-depth of the quantum AND gate is 1 Huang and Sun [27]. The quantum AND gate, together with its adjoint, are illustrated in Figure 1.

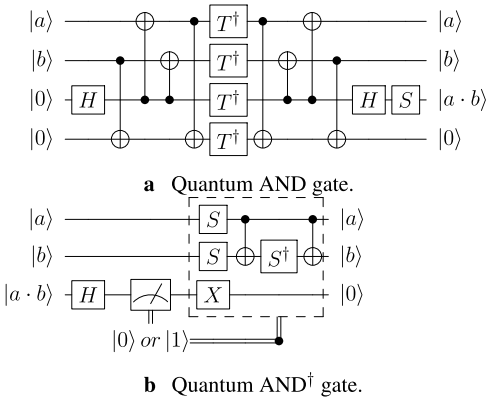

FIGURE 1
The quantum AND gate and its adjoint **(a)** quantum AND gate and **(b)** quantum AND† gate.

2.2 Classical circuits and directed acyclic graphs (DAG)

This paper utilizes concepts related to classical circuits and directed acyclic graphs (DAGs) to explain the circuit construction method Cong and Ding [39]. Therefore, this section will introduce these relevant concepts. A classical circuit can be represented by a DAG $C = (V, E)$ with a set of nodes V and a set of edges E. Nodes represent basic gates in classical circuits, and edges denote the direction of bit-flow in the classical circuit. The inputs of a classical circuit are often called the primary inputs. If there is a path from node v to node w, v is a predecessor of w, and w is a successor of v. Let us define the function $l(v) \rightarrow \{0,1\}$, which returns 1 if v is a multiplicative node and 0 otherwise. The multiplicative depth of node v is the maximum number of multiplicative gates on any path that begin with a primary input and end with a node v. The function of multiplicative depth d is

$$d(v) = \begin{cases} 0 & \text{if} |pred(v)| = 0 \\ \max_{u \in pred(v)} d(u) + l(v) & \text{otherwise} \end{cases}, \quad (1)$$

where $pred(v)$ denotes the set of all predecessor nodes of v. The multiplicative depth of a circuit C is the maximal multiplicative depth of its nodes

$$D = \max_{v \in C} d(v). \quad (2)$$

2.3 The advanced encryption standard

The advanced encryption standard (AES) is a block cipher standardized by the NIST Daemen [21]. Three variants, AES-128, AES-192, and AES-256, correspond to three different original key lengths. The detailed process of the AES algorithm is that the input 128-bit data block is initially XOR with the first 128 bits of the original key. Then, a specific number of round function iterations constitute the encryption process of AES: 10 rounds, 12 rounds, and 14 rounds for AES-128, AES-192, and AES-256, respectively. There are four operations during a round function in Figure 2: SubByte, ShiftRow, MixColumn, and AddRoundKey. MixColumn is omitted in the final round iteration.

Each block in a 4×4 matrix represents a byte. AddRoundKey applies the XOR operations to the round key and the 16 bytes. SubByte transforms the 16-byte state using the S-box. ShiftRow performs a cyclical leftward shift of the blocks in the i-th row by i positions, where $i = 0, 1, 2, 3$. MixColumn treats each column as a polynomial over the finite field $\mathbb{F}_{256}[x]/(x^4 + 1)$ and multiplies these polynomials by a fixed polynomial. It is a linear transformation that

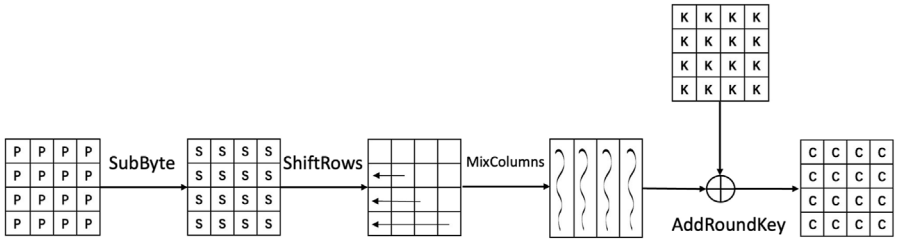

FIGURE 2
Round function process of the AES.

can be modeled as a matrix M in $\mathbb{F}_2^{32\times32}$

$$M = \begin{pmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x01 \end{pmatrix},$$

where the elements in matrix M are represented in hexadecimal. The current round key is generated from the preceding round key by Keyexpansion. Four bytes in a column of the 4×4 matrix represent a word W. Three operations are involved in the Keyexpansion: RotWord, Rcon, and SubWord. RotWord performs a cyclical leftward shift of words from the round key by one position. Rcon applies the XOR operations to the round key and a constant vector. SubWord transforms the state of one word from a round key by an S-box.

3 Constructing a circuit for an S-box with a low TDW and FDW

The universal method for constructing an S-box quantum circuit can be summarized in two steps. First, identify a classical circuit that implements the S-box. Then, implement this classical circuit in a quantum circuit. We chose the S-box classical circuit proposed by Boyar and Peralta [36], whose circuit depth had been optimized by heuristic algorithms. This section introduces the construction method: the controlled control qubit cascade (CCQC) technique for the S-box quantum circuit.

3.1 Constructing a quantum circuit with a low TDW and FDW

We introduce constructing an S-box quantum circuit by studying the S-box classical circuit Boyar and Peralta [36]. Based on the relevant knowledge presented in Equations 1, 2, we compute the multiplicative depth of all multiplicative nodes within this classical circuit and stratify all multiplicative nodes according to their multiplicative depth, as shown in Table 1, where we assign nodes with greater multiplicative depth to the higher layers.

The variables $U0, U1, \ldots, U7$ in the classical circuit Boyar and Peralta [36] represent the primary inputs of the S-box, while $S0, S1, \ldots, S7$ denote the outputs of the S-box. Analyzing these eight outputs leads to Observation 1. Based on our previous experience, the linear combinations of existing quantum states can be prepared easily by several CNOT gates in a quantum circuit. Therefore, our CCQC technique focuses on how to prepare multiplicative nodes with maximum multiplicative depth in classical circuits.

TABLE 1 Hierarchical multiplicative nodes from an S-box classical circuit.

Layer and multiplicative depth	Multiplicative node
1	M1, M2, M4, M6, M7, M9, M11, M12, and M14
2	M25, M31, and M34
3	M29, M30, M32, and M35
4	M46, M47, M48, M49, M50, M51, M52, M53, M54, M55, M56, M57, M58, M59, M60, M61, M62, and M63

Observation 1: *The eight outputs of the S-box can be derived by linear combinations of all multiplicative nodes with a multiplicative depth of 4.*

Example 1: $A1 = I1 \times I2$, $A2 = I2 \times I3$, $B1 = A1 + I3$, $B2 = A2 + I1$, $A3 = B2 \times I2$, $A4 = B2 \times I3$, $B3 = A3 + B1$, $B4 = B3 + A4$, $A5 = B2 \times B1$, $B5 = A5 + I2$, $A6 = B4 \times I3$, $A7 = B5 \times I2$.

We illustrate this method through a classical circuit in Example 1. We calculate the multiplicative depth of all multiplicative nodes in Example 1 and stratify them hierarchically. During the stratifying of multiplicative nodes in Example 1, we further observe that there usually exist additive nodes among different layers. These additive nodes merge nodes with smaller multiplicative depth to become the input for multiplicative nodes in a higher layer.

Theorem 1: *In a given classical circuit, any multiplicative node can be expressed as the product of a linear combination of its lower layer multiplicative nodes and the primary inputs.*

Theorem 1 shows that the highest-level multiplicative nodes can be obtained from lower-level multiplicative nodes and the primary inputs. The proof of Theorem 1 can be found in Supplementary Material. Table 2 lists the algebraic relationships between the multiplicative nodes from different layers of

TABLE 2 Algebraic relationships between the multiplicative nodes from Example 1.

Layer	Algebraic relationships
1	$A1 = I1 \times I2$, $A2 = I2 \times I3$
2	$A3 = (A2 + I1) \times I2$, $A4 = (A2 + I1) \times I3$, $A5 = (A2 + I1) \times (A1 + I3)$
3	$A6 = (A1 + A3 + A4 + I3) \times I3$, $A7 = (A5 + I2) \times I2$

```
Input: Any classical circuit C
Output: A novel classical circuit C that derived the algebraic relationships between multiplicative nodes
Function EXPAND(v)
  //v is the start node
i = = 1
for i ≤ 2 do
    //Input_v^i ∈ {Input_v^1, Input_v^2}
   if Input_v^i is addictive node then
      //Input_v^i is the input of v
    EXPAND(Input_v^i)
   else
    Update v with the production of Input_v^i in C
   end if
   i++
end for
End Function
for each node c ∈ C do
 if c is multiplicative node then
   EXPAND(c)
 else
   Continue
 end if
end for
Return C
```

Algorithm 1. Deriving algebraic relationships between multiplicative nodes in circuit $\mathcal{C}$.

Example 1. Algorithm 1 can ascertain the algebraic relationships between multiplicative nodes in any classical circuit.

The CCQC technique prepares multiplicative nodes in the quantum circuit hierarchically, and multiplicative nodes at the same layer are prepared in parallel. Because a qubit cannot be used in different quantum gates at the same time, to support the parallelism of quantum AND gates, ancilla qubits are needed to copy variables that are used as inputs to different multiplicative nodes. We can liken the inputs of multiplicative nodes to the control qubits of Toffoli gates and the outputs of the multiplicative nodes to controlled qubits. We use Toffoli gates due to the compactness of its visualization and the convenience of explanation. Then, the core concept of the CCQC technique is to use the controlled qubits of lower-layer multiplicative nodes as the control qubits for higher-layer multiplicative nodes. To elaborate, we take advantage of the qubits occupied by lower-layer

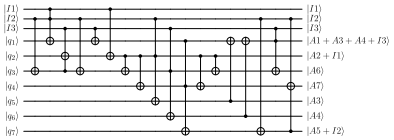

FIGURE 3
Constructing a quantum circuit with the CCQC technique.

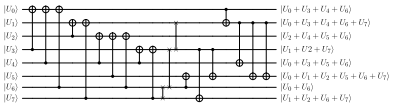

FIGURE 4
The quantum circuit to change the basis for the S-box.

multiplicative nodes as controlled qubits within the CNOT network, transforming these lower-layer controlled qubits into inputs for higher-layer multiplicative nodes. Figure 3 implements the highest-layer multiplicative nodes from Table 2 by our CCQC technique.

Constructing an S-box quantum circuit using the CCQC technique is more complex. We preprocess the S-box classical circuit followed by Theorem 1 and list the algebraic relationships among multiplicative nodes in Supplementary Material. Upon analyzing the classical circuit for the S-box, we obtain Observation 2.

Observation 2: *All inputs of the lowest layer multiplicative nodes are also the inputs of the highest layer multiplicative nodes.*

We find that it is beneficial to save qubits if primary inputs can be the inputs of multiplicative nodes directly. If not, extra qubits are needed to prepare inputs for multiplicative nodes. Therefore, we select eight linearly independent inputs $U_0 + U_3 + U_4 + U_6, U_0 + U_3 + U_4 + U_6 + U_7, U_2 + U_4 + U_5 + U_6, U_1 + U_2 + U_7, U_0 + U_3 + U_5 + U_6, U_0 + U_1 + U_2 + U_5 + U_6 + U_7, U_0 + U_6, U_1 + U_2 + U_6 + U_7$ from the first layer as a new set of basis. We adopt the method proposed by Patel et al. Markov et al. [40] to construct an in-place circuit that needs 17 CNOT gates to implement the transformation; see Figure 4. This new basis is used as the primary input when constructing an S-box quantum circuit with the help of Algorithm 2.

Algorithm 2 interprets how to construct a quantum circuit with the CCQC technique. List *Prepared* includes the prepared multiplicative nodes, and *Anc* includes ancilla qubits that are introduced to ensure quantum AND gates can be executed in parallel. Operation CNOT(m) means qubit m is a controlled qubit in the CNOT network, and Uncompute(q) means do an uncompute operation to release qubit q. Note that after preparing multiplicative nodes from the fourth layer, we still need to construct the outputs $|S0\rangle$, $|S1\rangle$,…, $|S7\rangle$ through CNOT gates and X gates. We list all allocated qubits and gates of the S-box quantum circuit constructed by Algorithm 2 in Table 3. Among them, the definitions of a CNOT gate, AND gate, X gate, and REWIRE operation are as follows: CNOT$a, b \to a, a + b$, AND$a, b, c \to a, b, c = a \cdot b$, X$(a) \to (a + 1)$, REWIRE$a, b \to b, a$. We implemented our S-box quantum circuit on Microsoft Q# to verify its correctness. The details can be viewed

TABLE 3 Quantum circuit for the S-box implementation.

No.	Gate	No.	Gate	No.	Gate
1	CNOTu_3, u_0	2	CNOTu_4, u_0	3	CNOTu_6, u_0
4	CNOTu_6, u_0	5	CNOTu_2, u_1	6	CNOTu_7, u_1
7	CNOTu_4, u_2	8	CNOTu_5, u_2	9	CNOTu_6, u_2
10	CNOTu_4, u_3	11	CNOTu_7, u_3	12	CNOTu_5, u_4
13	CNOTu_7, u_4	-	REWIREu_6, u_7	-	REWIREu_3, u_6
14	CNOTu_6, u_5	-	REWIREu_3, u_1	15	CNOTu_3, u_7
16	CNOTu_3, u_7	17	CNOTu_3, u_5	18	CNOTu_0, u_1
19	CNOTu_1, u_4	20	CNOTu_1, u_5	21	CNOTu_1, u_6
22	CNOTu_4, Anc_0	23	CNOTu_5, Anc_0	24	CNOTu_0, Anc_1
25	CNOTu_1, Anc_1	26	CNOTu_6, Anc_2	27	CNOTu_7, Anc_2
28	CNOTu_2, Anc_3	29	CNOTu_3, Anc_3	30	CNOTAnc_1, Anc_4
31	CNOTAnc_3, Anc_4	32	CNOTu_4, Anc_5	33	CNOTu_6, Anc_5
34	CNOTu_1, Anc_6	35	CNOTu_3, Anc_6	36	CNOTu_5, Anc_7
37	CNOTu_7, Anc_7	38	CNOTAnc_6, Anc_8	39	CNOTAnc_4, Anc_8
40	CNOTu_4, Anc_9	41	CNOTu_6, Anc_9	42	CNOTAnc_7, Anc_9
43	ANDu_4, u_0, Anc_{10}	44	ANDu_5, u_1, Anc_{11}	45	ANDAnc_0, Anc_1, Anc_{12}
46	ANDu_6, u_2, Anc_{13}	47	ANDu_7, u_3, Anc_{14}	48	ANDAnc_2, Anc_3, Anc_{15}
49	ANDAnc_9, Anc_4, Anc_{16}	50	ANDAnc_5, Anc_8, Anc_{17}	51	ANDAnc_7, Anc_6, Anc_{18}
52	CNOTu_6, Anc_{19}	53	CNOTu_2, Anc_{19}	54	CNOTAnc_{13}, Anc_{19}
55	CNOTAnc_{14}, Anc_{19}	56	CNOTAnc_{16}, Anc_{19}	57	CNOTAnc_{17}, Anc_{19}
58	CNOTAnc_9, Anc_{20}	59	CNOTAnc_4, Anc_{20}	60	CNOTAnc_{10}, Anc_{20}
61	CNOTAnc_{11}, Anc_{20}	62	CNOTAnc_{16}, Anc_{20}	63	CNOTAnc_{17}, Anc_{20}
64	CNOTAnc_2, Anc_{21}	65	CNOTAnc_3, Anc_{21}	66	CNOTAnc_{13}, Anc_{21}
67	CNOTAnc_{15}, Anc_{21}	68	CNOTAnc_{16}, Anc_{21}	69	CNOTAnc_{18}, Anc_{21}
70	CNOTAnc_7, Anc_{22}	71	CNOTAnc_6, Anc_{22}	72	CNOTAnc_{10}, Anc_{22}
73	CNOTAnc_{12}, Anc_{22}	74	CNOTAnc_{16}, Anc_{22}	75	CNOTAnc_{18}, Anc_{22}
76	CNOTAnc_{19}, Anc_{23}	77	CNOTAnc_{20}, Anc_{24}	78	ANDAnc_{19}, Anc_{20}, Anc_{25}
79	ANDAnc_{24}, Anc_{21}, Anc_{26}	80	ANDAnc_{22}, Anc_{23}, Anc_{27}	81	CNOTAnc_{25}, Anc_{21}
82	CNOTAnc_5, Anc_{28}	83	CNOTAnc_8, Anc_{28}	84	CNOTAnc_{11}, Anc_{28}
85	CNOTAnc_{12}, Anc_{28}	86	CNOTAnc_{17}, Anc_{28}	87	CNOTAnc_{18}, Anc_{28}
88	CNOTAnc_{25}, Anc_{22}	89	CNOTu_7, Anc_{29}	90	CNOTu_3, Anc_{29}
91	CNOTAnc_{14}, Anc_{29}	92	CNOTAnc_{15}, Anc_{29}	93	CNOTAnc_{17}, Anc_{29}

(Continued on the following page)

TABLE 3 (*Continued*) Quantum circuit for the S-box implementation.

No.	Gate	No.	Gate	No.	Gate
94	CNOTAnc_{18}, Anc_{29}	95	CNOTAnc_{28}, Anc_{30}	96	CNOTAnc_{29}, Anc_{31}
97	ANDAnc_{21}, Anc_{28}, Anc_{32}	98	ANDAnc_{22}, Anc_{29}, Anc_{33}	99	ANDAnc_{30}, Anc_{26}, Anc_{34}
100	ANDAnc_{31}, Anc_{27}, Anc_{35}	101	CNOTAnc_{31}, Anc_{29}	102	CNOTAnc_{24}, Anc_{20}
103	CNOTAnc_{30}, Anc_{28}	104	CNOTAnc_{19}, Anc_{21}	105	CNOTAnc_{31}, Anc_{21}
106	CNOTAnc_{25}, Anc_{21}	107	CNOTAnc_{25}, Anc_{19}	108	CNOTAnc_{33}, Anc_{19}
109	CNOTAnc_{35}, Anc_{19}	110	CNOTAnc_{25}, Anc_{22}	111	CNOT(Anc_{32},Anc_{22})
112	CNOT(Anc_{25},Anc_{23})	113	CNOT(Anc_{33},Anc_{23})	114	CNOT(Anc_{35},Anc_{23})
115	CNOT(Anc_{25},Anc_{31})	116	CNOT(Anc_{35},Anc_{31})	117	CNOT(Anc_{25},Anc_{24})
118	CNOT(Anc_{32},Anc_{24})	119	CNOT(Anc_{34},Anc_{24})	120	CNOT(Anc_{25},Anc_{30})
121	CNOT(Anc_{34},Anc_{30})	122	CNOT(Anc_{21},Anc_{29})	123	CNOT(Anc_{22},Anc_{29})
124	CNOT(Anc_{29},Anc_{36})	125	CNOT(Anc_{29},Anc_{36})	126	CNOT(Anc_{19},Anc_{20})
127	CNOT(Anc_{24},Anc_{20})	128	CNOT(Anc_{20},Anc_{37})	129	CNOT(Anc_{30},Anc_{28})
130	CNOT(Anc_{31},Anc_{28})	131	CNOT(Anc_{28},Anc_{21})	132	CNOT(Anc_{15},Anc_{13})
133	CNOT(Anc_{16},Anc_{13})	134	CNOT(Anc_{18},Anc_{13})	135	CNOT(Anc_{33},Anc_{13})
136	CNOT(u_0,Anc_{13})	137	CNOT(u_2,Anc_{13})	138	CNOT(u_4,Anc_{13})
139	CNOTAnc_{13}, Anc_{38}	140	CNOTAnc_{22}, Anc_{39}	141	CNOTAnc_{11}, Anc_{10}
142	CNOTAnc_{16}, Anc_{10}	143	CNOTAnc_{17}, Anc_{10}	144	CNOTAnc_{25}, Anc_{10}
145	CNOTAnc_{32}, Anc_{10}	146	CNOTAnc_{34}, Anc_{10}	147	CNOTu_0, Anc_{10}
148	CNOTu_2, Anc_{10}	149	CNOTu_3, Anc_{10}	150	CNOTu_4, Anc_{10}
151	CNOTu_5, Anc_{10}	152	CNOTAnc_{12}, Anc_{11}	153	CNOTAnc_{17}, Anc_{11}
154	CNOTAnc_{18}, Anc_{11}	155	CNOTAnc_{25}, Anc_{11}	156	CNOTAnc_{34}, Anc_{11}
157	CNOTu_0, Anc_{11}	158	CNOTu_2, Anc_{11}	159	CNOTu_4, Anc_{11}
160	CNOTu_6, Anc_{11}	161	CNOTAnc_{15}, Anc_{14}	162	CNOTAnc_{17}, Anc_{14}
163	CNOTAnc_{18}, Anc_{14}	164	CNOT(Anc_{25},Anc_{14})	165	CNOTAnc_{35}, Anc_{14}
166	CNOTu_2, Anc_{14}	167	CNOTu_6, Anc_{14}	168	ANDAnc_{19}, u_0, Anc_{40}
169	ANDAnc_{14}, u_1, Anc_{41}	170	ANDAnc_{13}, Anc_1, Anc_{42}	171	ANDAnc_{10}, u_2, Anc_{43}
172	ANDAnc_{11}, u_3, Anc_{44}	173	ANDAnc_{22}, Anc_3, Anc_{45}	174	ANDAnc_{29}, Anc_4, Anc_{46}
175	ANDAnc_{20}, Anc_8, Anc_{47}	176	ANDAnc_{28}, Anc_6, Anc_{48}	177	ANDAnc_{23}, u_4, Anc_{49}
178	ANDAnc_{31}, u_5, Anc_{50}	179	ANDAnc_{38}, Anc_0, Anc_{51}	180	ANDAnc_{24}, u_6, Anc_{52}
181	ANDAnc_{30}, u_7, Anc_{53}	182	ANDAnc_{39}, Anc_2, Anc_{54}	183	ANDAnc_{36}, Anc_9, Anc_{55}
184	ANDAnc_{37}, Anc_5, Anc_{56}	185	ANDAnc_{21}, Anc_7, Anc_{58}	186	CNOTAnc_{43}, s_0
187	CNOTAnc_{44}, s_0	188	CNOTAnc_{46}, s_0	189	CNOTAnc_{20}, s_0

(Continued on the following page)

TABLE 3 (*Continued*) Quantum circuit for the S-box implementation.

No.	Gate	No.	Gate	No.	Gate
190	CNOTAnc_{49}, s_0	191	CNOTAnc_{50}, s_0	192	CNOTAnc_{55}, s_0
193	CNOTAnc_{56}, s_0	194	CNOTAnc_{40}, s_1	195	CNOTAnc_{41}, s_1
196	CNOTAnc_{46}, s_1	197	CNOTAnc_{47}, s_1	198	CNOTAnc_{49}, s_1
199	CNOTAnc_{50}, s_1	200	CNOTAnc_{55}, s_1	201	CNOTAnc_{56}, s_1
202	X(s_1)	203	CNOTAnc_{40}, s_2	204	CNOTAnc_{42}, s_2
205	CNOTAnc_{46}, s_2	206	CNOTAnc_{48}, s_2	207	CNOTAnc_{52}, s_2
208	CNOTAnc_{54}, s_2	210	CNOTAnc_{55}, s_2	211	CNOTAnc_{58}, s_2
212	X(s_2)	213	CNOTAnc_{40}, s_3	213	CNOTAnc_{41}, s_3
214	CNOTAnc_{43}, s_3	215	CNOTAnc_{44}, s_3	216	CNOTAnc_{49}, s_3
217	CNOTAnc_{50}, s_3	218	CNOTAnc_{55}, s_3	219	CNOTAnc_{56}, s_3
220	CNOTAnc_{41}, s_4	221	CNOTAnc_{42}, s_4	222	CNOTAnc_{44}, s_4
223	CNOTAnc_{45}, s_4	224	CNOTAnc_{49}, s_4	225	CNOTAnc_{50}, s_4
226	CNOTAnc_{55}, s_4	227	CNOTAnc_{56}, s_4	228	CNOTAnc_{40}, s_5
229	CNOTAnc_{42}, s_5	230	CNOTAnc_{43}, s_5	231	CNOTAnc_{44}, s_5
232	CNOTAnc_{47}, s_5	233	CNOTAnc_{48}, s_5	234	CNOTAnc_{50}, s_5
235	CNOTAnc_{51}, s_5	236	CNOTAnc_{52}, s_5	237	CNOTAnc_{54}, s_5
238	CNOTAnc_{55}, s_5	239	CNOTAnc_{56}, s_5	240	CNOTAnc_{44}, s_6
241	CNOTAnc_{45}, s_6	242	CNOTAnc_{47}, s_6	243	CNOTAnc_{48}, s_6
244	CNOTAnc_{52}, s_6	245	CNOTAnc_{53}, s_6	246	CNOTAnc_{55}, s_6
247	CNOTAnc_{56}, s_6	248	X(s_6)	249	CNOTAnc_{40}, s_7
250	CNOTAnc_{42}, s_7	251	CNOTAnc_{43}, s_7	252	CNOTAnc_{45}, s_7
253	CNOTAnc_{52}, s_7	254	CNOTAnc_{53}, s_7	255	CNOTAnc_{55}, s_7
256	CNOTAnc_{56}, s_7	257	X(s_7)		

in the online code at https://github.com/kyolxs/Constructing-Resource-Efficient-Quantum-Circuits-for-AES.

Various implementations of S-box quantum circuits have been proposed, with some utilizing Toffoli gates and others employing AND gates. We present a comparison of the quantum resource costs in S-box quantum circuits based on the Toffoli gate in Table 4, focusing on previous works that study low Toffoli depth S-boxes. Additionally, Jang et al. [35] and Liu et al. [37] conducted a comprehensive comparison by decomposing Toffoli gates in different manners. However, this paper adopts AND gates to achieve lower circuit depth. Therefore, we need an additional 18 ancilla qubits to support the preparation of the fourth layer multiplicative nodes. After executing the fourth layer of AND gates, those 18 ancilla bits are reset to $|0\rangle$, and eight of them can be used directly to prepare $|S0\rangle$, $|S1\rangle$,…, $|S7\rangle$. After the execution of the S-box, an S-box† is necessary to uncompute ancilla qubits. We list a comparison of the quantum resources costs in the S-box and the S-box† based on the AND gate in Table 5. Among those works, the S-box quantum circuit proposed by Liu et al. at ASIACRYPT 2023 Liu et al. [37] has the same T-depth and a similar width as our work. Because their proposed m-XOR technique can identify reusable qubits, it is very effective in reducing width. Therefore, their work is similar to our work in terms of the TDW. In fact, during our preprocessing to obtain an S-box classical circuit, we frequently find that variables M_j or U_i appear even times within the same formulation $\sum_j x_j M_j + \sum_i y_i U_i, x_j, y_i \in \mathbb{F}_2$. Due to the properties of the XOR operation, these variables can be directly eliminated, avoiding unnecessary CNOT gates targeting the same qubit and thereby reducing the F-depth.

TABLE 4 Comparison of quantum resources for low Toffoli depth S-boxes based on a Toffoli gate.

Source	Ancilla qubits	#Toffoli	*To f*-depth	#CNOT	#1qClifford
[26]	120	34	6	186	4
[27]	120	34	4	214	4
[27]	182	78	3	356	4
[42]	136	78	3	313	4
[42]	68	34	4	162	4
[37]	74	34	4	168	4
[37]	60	34	4	196	4
[43]	74	34	4	179	4
[43]	60	34	4	207	4
Ours	58	34	4	219	4

The bold values indicate the quantum resources required by the S-box quantum circuit constructed in this work.

TABLE 5 Comparison of quantum resources for an S-box and an S-box† based on an AND gate.

Source	#CNOT	#1qClifford	#T	T-depth	F-depth	Width	Depth × width	
							TDW	FDW
[26]	664	205	136	6	117	136	816	15,912
[27]	718	208	136	4	109	136	544	14,824
[37]	624	204	136	4	101	99	396	9,999
[37]	688	220	136	4	132	86	344	11,352
Our	710	276	136	4	74	58 + 8 + 18 = 84	**336**	**6,216**

The bold values indicate the TDW and FDW of the S-box quantum circuit constructed in this work.

Two types of S-boxes are required to construct the AES quantum circuit in the next section. The first S-box implements $|x\rangle^{\otimes 8}|0\rangle^{\otimes 8}|0\rangle^{\otimes a} \rightarrow |x\rangle^{\otimes 8}|S(x)\rangle^{\otimes 8}|0\rangle^{\otimes a}$ in SubByte. The second S-box implements $|x\rangle^{\otimes 8}|y\rangle^{\otimes 8}|0\rangle^{\otimes a} \rightarrow |x\rangle^{\otimes 8}|y + S(x)\rangle^{\otimes 8}|0\rangle^{\otimes a}$ in SubWord. Our S-box quantum circuit can directly accommodate both situations. We do not differentiate between these two types of S-boxes in our subsequent discussions.

4 Optimized quantum circuits for AES

This section discusses the implementation of the quantum circuit of AES. We begin by explaining how each component can be implemented in a quantum circuit, followed by how to implement the iterative encryption circuits for AES under a pipeline structure and a round-in-place structure. We address estimating the quantum resources required to implement AES. It is important to emphasize that all the circuits mentioned in this section are implemented with the maximum parallelism numbers of an S-box.

4.1 Components of AES and their implementations

4.1.1 SubByte and SubWord

The S-box is the core cryptographic component used to implement SubByte and SubWord. Its quantum circuit implementation is detailed in previous sections. SubByte needs 16 S-boxes, and SubWord needs four S-boxes.

4.1.2 ShiftRow and RotWord

ShiftRow and RotWord only perform cyclical leftward shifts but do not change the state of bytes. Both can be implemented in a quantum circuit entirely through rewiring. Following Grassl et al. [23], we considered rewiring as a free operation, thus excluding it from cost estimates.

4.1.3 MixColumn and Rcon

MixColumn can be implemented with an in-place quantum circuit due to the invertibility of its 32×32 binary matrix M. A

TABLE 6 The key schedule process of AES-192.

Round	$\|K_0\rangle^{\otimes 32}$	$\|K_1\rangle^{\otimes 32}$	$\|K_2\rangle^{\otimes 32}$	$\|K_3\rangle^{\otimes 32}$	$\|K_4\rangle^{\otimes 32}$	$\|K_5\rangle^{\otimes 32}$
0	$\mathbf{W_0}$	$\mathbf{W_1}$	$\mathbf{W_2}$	$\mathbf{W_3}$	W_4	W_5
1	$\widetilde{\mathbf{W_6}}$	$\mathbf{W_7}$	W_8	W_9	$\mathbf{W_4}$	$\mathbf{W_5}$
2	W_6	W_7	$\mathbf{W_8}$	$\mathbf{W_9}$	$\mathbf{W_{10}}$	$\mathbf{W_{11}}$
3	$\widetilde{\mathbf{W_{12}}}$	$\mathbf{W_{13}}$	$\mathbf{W_{14}}$	$\mathbf{W_{15}}$	W_{16}	W_{17}
4	$\widetilde{\mathbf{W_{18}}}$	$\mathbf{W_{19}}$	W_{14}	W_{15}	$\mathbf{W_{16}}$	$\mathbf{W_{17}}$
5	W_{18}	W_{19}	$\mathbf{W_{20}}$	$\mathbf{W_{21}}$	$\mathbf{W_{22}}$	$\mathbf{W_{23}}$
6	$\widetilde{\mathbf{W_{24}}}$	$\mathbf{W_{25}}$	$\mathbf{W_{26}}$	$\mathbf{W_{27}}$	W_{28}	W_{29}
7	$\widetilde{\mathbf{W_{30}}}$	$\mathbf{W_{31}}$	W_{26}	W_{27}	$\mathbf{W_{28}}$	$\mathbf{W_{29}}$
8	W_{30}	W_{31}	$\mathbf{W_{32}}$	$\mathbf{W_{33}}$	$\mathbf{W_{34}}$	$\mathbf{W_{35}}$
9	$\widetilde{\mathbf{W_{36}}}$	$\mathbf{W_{37}}$	$\mathbf{W_{38}}$	$\mathbf{W_{39}}$	W_{40}	W_{41}
10	$\widetilde{\mathbf{W_{42}}}$	$\mathbf{W_{43}}$	W_{38}	W_{39}	$\mathbf{W_{40}}$	$\mathbf{W_{41}}$
11	W_{42}	W_{43}	$\mathbf{W_{44}}$	$\mathbf{W_{45}}$	$\mathbf{W_{46}}$	$\mathbf{W_{47}}$
12	$\widetilde{\mathbf{W_{48}}}$	$\mathbf{W_{49}}$	$\mathbf{W_{50}}$	$\mathbf{W_{51}}$	W_{46}	W_{47}

The bold keywords denote the keywords used in the current round of iteration.

TABLE 7 The key schedule process of AES-256.

Round	$\|K_0\rangle^{\otimes 32}$	$\|K_1\rangle^{\otimes 32}$	$\|K_2\rangle^{\otimes 32}$	$\|K_3\rangle^{\otimes 32}$	$\|K_4\rangle^{\otimes 32}$	$\|K_5\rangle^{\otimes 32}$	$\|K_6\rangle^{\otimes 32}$	$\|K_7\rangle^{\otimes 32}$
0	$\mathbf{W_0}$	$\mathbf{W_1}$	$\mathbf{W_2}$	$\mathbf{W_3}$	W_4	W_5	W_6	W_7
1	$\widetilde{W_8}$	W_9	W_{10}	W_{11}	$\mathbf{W_4}$	$\mathbf{W_5}$	$\mathbf{W_6}$	$\mathbf{W_7}$
2	$\mathbf{W_8}$	$\mathbf{W_9}$	$\mathbf{W_{10}}$	$\mathbf{W_{11}}$	W_{12}	W_{13}	W_{14}	W_{15}
3	$\widetilde{W_{16}}$	W_{17}	W_{18}	W_{19}	$\mathbf{W_{12}}$	$\mathbf{W_{13}}$	$\mathbf{W_{14}}$	$\mathbf{W_{15}}$
4	$\mathbf{W_{16}}$	$\mathbf{W_{17}}$	$\mathbf{W_{18}}$	$\mathbf{W_{19}}$	W_{20}	W_{21}	W_{22}	W_{23}
5	$\widetilde{W_{24}}$	W_{25}	W_{26}	W_{27}	$\mathbf{W_{20}}$	$\mathbf{W_{21}}$	$\mathbf{W_{22}}$	$\mathbf{W_{23}}$
6	$\mathbf{W_{24}}$	$\mathbf{W_{25}}$	$\mathbf{W_{26}}$	$\mathbf{W_{27}}$	W_{28}	W_{29}	W_{30}	W_{31}
7	$\widetilde{W_{32}}$	W_{33}	W_{34}	W_{35}	$\mathbf{W_{28}}$	$\mathbf{W_{29}}$	$\mathbf{W_{30}}$	$\mathbf{W_{31}}$
8	$\mathbf{W_{32}}$	$\mathbf{W_{33}}$	$\mathbf{W_{34}}$	$\mathbf{W_{35}}$	W_{36}	W_{37}	W_{38}	W_{39}
9	$\widetilde{W_{40}}$	W_{41}	W_{42}	W_{43}	$\mathbf{W_{36}}$	$\mathbf{W_{37}}$	$\mathbf{W_{38}}$	$\mathbf{W_{39}}$
10	$\mathbf{W_{40}}$	$\mathbf{W_{41}}$	$\mathbf{W_{42}}$	$\mathbf{W_{43}}$	W_{44}	W_{45}	W_{46}	W_{47}
11	$\widetilde{W_{48}}$	W_{49}	W_{50}	W_{51}	$\mathbf{W_{44}}$	$\mathbf{W_{45}}$	$\mathbf{W_{46}}$	$\mathbf{W_{47}}$
12	$\mathbf{W_{48}}$	$\mathbf{W_{49}}$	$\mathbf{W_{50}}$	$\mathbf{W_{51}}$	W_{52}	W_{53}	W_{54}	W_{55}
13	$\widetilde{W_{56}}$	W_{57}	W_{58}	W_{59}	$\mathbf{W_{52}}$	$\mathbf{W_{53}}$	$\mathbf{W_{54}}$	$\mathbf{W_{55}}$
14	$\mathbf{W_{56}}$	$\mathbf{W_{57}}$	$\mathbf{W_{58}}$	$\mathbf{W_{59}}$	W_{52}	W_{53}	W_{54}	W_{55}

The bold keywords denote the keywords used in the current round of iteration.

TABLE 8 Quantum resources for implementing AES and its adjoint circuit with a pipeline structure.

	Source	#CNOT	#1qClifford	#T	T-depth	F-depth	Width	Depth × width	
								TDW	FDW
AES-128	[26]	291,150	83,116	54,400	120	2,827	3,936	472,320	11,127,072
	[27]	298,720	83,295	54,400	80	2,198	3,936	314,880	8,651,328
	[35]*	285,984	75,040	54,400	80	1,856	6,372	509,760	11,826,432
	[37]	264,752	103,600	54,400	80	1,600	3,689	295,120	5,902,400
	[27]	570,785	189,026	124,800	60	2,312	5,576	334,560	1,2,891,712
	Our	189,300	42,440	54,400	**80**	1,790	2,896	**231,680**	**5,183,840**
AES-192	[35]	325,072	86,384	60,928	96	2,228	6,692	642,432	14,909,776
	[37]	298,512	116,848	60,928	96	1924	3,945	378,720	7,590,180
	Our	212,652	47,520	60,928	**96**	2,154	3,216	**308,736**	**6,927,264**
AES-256	[35]	399,792	106,654	75,072	112	2,614	6,980	781,760	18,245,720
	[37]	374,256	147,758	38,080	112	2,240	4,457	499,184	9,983,680
	Our	261,148	58,564	75,072	**112**	2,518	3,536	**396,032**	**8,903,648**

The bold values indicate the T-depth, TDW, and FDW of the AES quantum circuit constructed in this work.

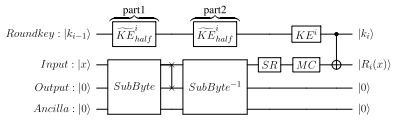

FIGURE 5
The iterative encryption circuit with a round-in-place structure.

number of studies by Jang et al. [35]; Liu et al. [37]; Xiang et al. [41] have been conducted on the implementation of MixColumn. We use an in-place circuit Liu et al. [37] that requires only 98 CNOT gates to implement MixColumn with an F-depth of 16. It can easily be realized for the Rcon operation by applying X gates in the first word of the round key as necessary.

4.1.4 AddRoundKey

AddRoundKey is executed in a quantum circuit through a CNOT network, where the round key qubits act as control bits and the 128 qubits for the AES data block act as controlled bits in the CNOT network.

4.1.5 Keyexpansion

As mentioned before, the key used in the AddRoundKey comes from the round key generated by the Keyexpansion. Hence, Keyexpansion is also an iterative process that uses ten rounds, eight rounds, and seven rounds for AES-128, AES-192, and AES-256, respectively. In this paper, we adopt the in-place circuit structure proposed by Jaques et al. [26] to realize the Keyexpansion iteration by combining it with the SubByte subcircuit implemented by S-boxes. For detailed circuit structure, please refer to Jaques et al. [26]. Each round of Keyexpansion process generates four, six, and eight words that correspond to AES-128, AES-192, and AES-256, respectively.

The key schedule strategy controls the iterative progress of Keyexpansion, and a reasonable key schedule strategy can make the circuit more compact. We designed a new key schedule strategy that reduces the F-depth. The criterion of our key schedule strategy is to ensure that the SubWord subcircuit must be executed in parallel with the SubByte subcircuit. For AES-128, the Keyexpansion's iteration is synchronized with the round function's iteration. However, it should be noted that subcircuits of Rcon and SubWord must be executed in parallel with the SubByte subcircuit, and the remainder subcircuit of Keyexpansion executes in parallel with the MixColumn subcircuit. Due to the iterative rounds of Keyexpansion, it is slightly more complex to arrange the circuit layout of AES-192 and AES-256 in a reasonable way to make the circuit more compact. We summarize the key schedule process corresponding to each iteration of the round function for AES-192 and AES-256 in Tables 6, 7. The leftmost column in the tables indicates the round function's iterative round, while the remaining columns show which key word is stored in the 32 qubit registers during the key schedule process. $W_0, W_1, \ldots, W_5$ are original key of AES-192, and $W_0, W_1, \ldots, W_7$ are original key of AES-256. The key words with a wavy line on top indicate the key words generated by SubWord and need to be executed in parallel with the SubByte subcircuit. The bold keywords denote the key words used in the current round of iteration.

TABLE 9 Quantum resources for implementing forward AES with a round-in-place structure.

	Source	T-depth	F-depth	Width	Depth × width	
					TDW	FDW
AES-128	[27]†	2,460	—	492	1,210,320	—
	[37]	80	2,796	1,660	132,800	4,641,360
	Our	80	1,636	**1,608**	**128,640**	**2,630,688**
AES-192	Our	96	1966	1,672	160,512	3,287,152
AES-256	Our	112	2,296	1,736	194,432	3,985,856

The bold values indicate the Width, TDW, and FDW of the AES quantum circuit constructed in this work.

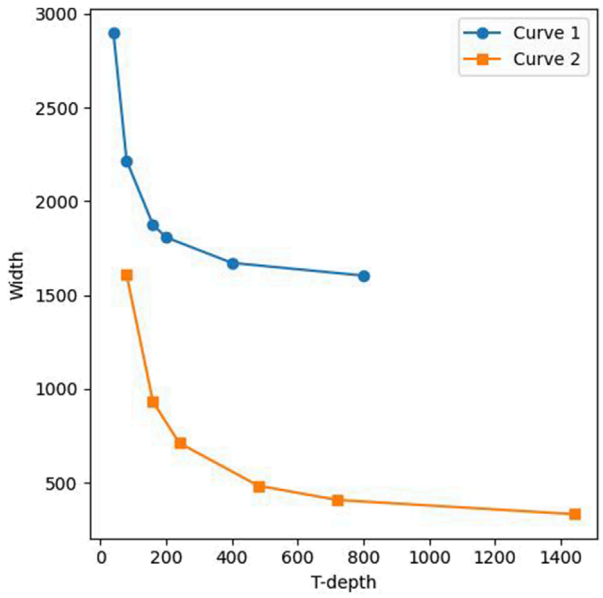

FIGURE 6
The trade-off between T-depth and width for AES-128.

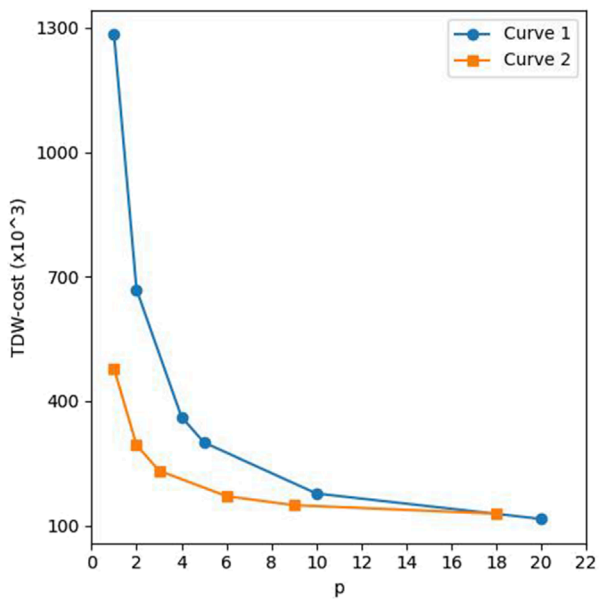

FIGURE 7
The influence of p on TDW.

4.2 AES quantum circuit with a pipeline structure

The pipeline structure Jaques et al. [26] was proposed by Jaques et al. to reduce the depth of the circuit. The characteristic of the pipeline structure is that after completing one round iterative round function, it directly allocates new qubits to implement the next round iterative round function. Jang et al. [42] further refined the pipeline structure into a regular version and a shallow version. We adopted the regular version because the regular version has high parallelism while considering the depth-qubit trade-off. We integrated our S-box quantum circuit into the regular-version pipeline structure and estimated the quantum resources required to implement the AES forward circuit and its adjoint circuit. Table 8 compares our work with previous works under the same structure. It is worth noting that Liu et al. [37] and Jang et al. [35] only provide the resources for a forward circuit. Therefore, we have multiplied the metrics other than width by 2 in Table 8.

For the implementation of AES-128, the quantum circuit that applies our S-box quantum circuit with pipeline structure has a T-depth of 80. Compared to the state-of-the-art work with the same T-depth, our approach achieves a 21.5% reduction in TDW and a 12.2% reduction in FDW. In the case of AES-192, the quantum circuit that applies our S-box quantum circuit with a pipeline structure has a T-depth of 96. Compared to the state-of-the-art work with the same T-depth, our approach achieves an 18.5% reduction in TDW and an 8.7% reduction in FDW. Regarding AES-256, the quantum circuit that applies our S-box quantum circuit with a pipeline structure has a T-depth of 112. Compared to the state-of-the-art work with the same T-depth, our approach achieves a 20.7% reduction in TDW and a 10.9% reduction in FDW.

Input: Algebraic relationships between multiplicative nodes of a given classical circuit
Output: A quantum circuit to implement the highest layer
Initialize empty lists *Prepared* and *Anc*
$l = 1$
//L denotes maximum layer of $\mathcal{C}$
while $l \le L$ **do**
if *Anc* is not empty **then**
for $q \in$ *Anc* **do**
if q is not required by l-th layer **then**
Uncompute(q)
remove q from *Anc*
end if
end for
end if
for $M_i \in l$-th layer **do**
if $\exists m \in$ *Prepared* is required by $M_i \wedge$ only required by M_i **then**
CNOT(m)
else
Introduce new ancilla qubit q_{new} and add q_{new} to *Anc*
CNOT(q_{new})
end if
end for
prepare multiplicative nodes in l-th layer by quantum AND gates
for $M_i \in l$-th layer **do**
add M_i to *Prepared*
end for
$l = l + 1$
end while

Algorithm 2. Apply the CCQC technique to construct the quantum circuit.

4.3 AES quantum circuit with a round-in-place structure

The round-in-place structure Huang and Sun [27] was proposed by Huang et al. to maximize the reuse of qubits and greatly reduce circuit width. They showed the method to construct the inverse S-box quantum circuit based on an S-box quantum circuit, an additional 42 CNOT gates, and four X gates. For more details, please refer to Huang and Sun [27]. With the inverse S-box quantum circuit, they constructed the iterative encryption in an in-place manner; see Figure 5. Because the T-depth of one round iteration with the round-in-place structure is twice that of the pipeline structure, we divide $\widetilde{KE}$ into two halves to save ancilla qubits. After dividing $\widetilde{KE}$ into two-halves, there needs an accumulative total of 18 S-boxes in each part, two S-boxes in $\widetilde{KE}_{half}$ and 16 S-boxes or inverse S-boxes in *SubByte* or $SubByte^{-1}$.

We also apply our S-box quantum circuit to the round-in-place structure and estimate quantum resources for implementing the AES forward circuit without its adjoint circuit. Table 9 compares our work with previous works under the round-in-place structure. Because Liu et al. [37] did not provide the F-depth of AES-128, we calculate the F-depth of their circuit based on the F-depth of the S-box they applied to an R structure. Apart from the F-depth, all other metrics are directly derived from Liu et al. The quantum circuit that applies our S-box quantum circuit with a round-in-place structure had a T-depth of 80. Compared with the circuit of Liu et al. with the same T-depth, we achieved a modest reduction of 3.1% in TDW and a substantial reduction of 43.3% in FDW.

5 Trade-offs of circuit metrics

This section shows the trade-offs between T-depth and width, the parallel numbers p of the S-box and TDW. We construct four AES-128 circuits by combining the pipeline structure and round-in-place structure with our S-box quantum circuit. We set different values for p in these four circuits and estimate their width, T-depth, and TDW. We define the pipeline structure using our S-box quantum circuit as Circuit 1, and the round-in-place structure using our S-box quantum circuit is represented by Circuit 2. It should be noted that due to the different structural characteristics of these two circuit structures, we set p for the pipeline structure to be a factor of 20 and p for the round-in-place structure to be a factor of 18. Figure 6 shows the trade-off curves for the T-depth and width of the different circuits. Figure 7 reflects the influence of p on TDW. The points on the curves correspond to different values of p. In the same curve, the points on the right correspond to smaller values of p. From left to right, the points on Circuit 1 correspond to $p = 20, 10, 5, 4, 2, 1$, respectively. From left to right, points on Circuit 2 correspond to $p = 18, 9, 6, 3, 2, 1$, respectively.

6 Conclusion

We propose the CCQC technique for constructing quantum circuits for nonlinear components, providing a general method to reduce both TDW and FDW. Additionally, we design a new key schedule strategy to reduce the F-depth of the AES quantum circuit. This paper introduces significantly optimized AES quantum circuits, achieving improvements in TDW and FDW. Our research provides a new idea for constructing quantum circuits for nonlinear components of other block ciphers with low TDW and FDW. However, the CCQC technique is more suitable for constructing quantum circuits with low TDW and FDW for classical circuits that have a lower multiplicative depth. On the other hand, solely focusing on reducing the multiplicative depth of a circuit implies the need for more intermediate values, which in turn costs more qubits when constructing the quantum circuit. Therefore, the impact of multiplicative depth on TDW and FDW is worth further exploration. Finally, we find that for the S-box quantum circuit designed in this article, implementing the iterative AES quantum

circuit by regular-version pipeline structure is more advantageous for FDW and TDW.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

L-LJ: conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, and writing – review and editing. B-BC: writing – original draft and writing – review and editing. FG: writing – original draft, writing – review and editing, conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, and visualization. S-JQ: writing – original draft and writing – review and editing. Z-PJ: writing – original draft and writing – review and editing. Q-YW: writing – original draft and writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work is supported by the National Natural Science Foundation of China (Grant Nos. 62272056, 62372048, and 62371069).

Acknowledgments

I would like to express my sincere gratitude to the Tianyan Quantum Computing Program for providing valuable resources and support. Your platform has greatly facilitated my learning and deepened my understanding of quantum computing.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2025.1582819/full#supplementary-material

References

1. Shor PW, Preskill J. Simple proof of security of the bb84 quantum key distribution protocol. *Phys Rev Lett* (2000) 85:441–4. doi:10.1103/physrevlett.85.441

2. Qin L, Liu B, Gao F, Huang W, Xu B, Li Y. Decoy-state quantum private query protocol with two-way communication. *Physica A: Stat Mech its Appl* (2024) 633:129427. doi:10.1016/j.physa.2023.129427

3. Grover LK. A fast quantum mechanical algorithm for database search. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996). p. 212–9.

4. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings 35th annual symposium on foundations of computer science (Ieee)* (1994). p. 124–34.

5. Braginsky VB, Khalili FY. *Quantum measurement*. Cambridge University Press (1995).

6. Childs AM, Preskill J, Renes J. Quantum information and precision measurement. *J Mod Opt* (2000) 47:155–76. doi:10.1080/095003400148123

7. Zhou N-R, Chen Z-Y, Liu Y-Y, Gong L-H. Multi-party semi-quantum private comparison protocol of size relation with d-level ghz states. *Adv Quan Tech* (2024):2400530. doi:10.1002/qute.202400530

8. Gong L-H, Li M-L, Cao H, Wang B. Novel semi-quantum private comparison protocol with bell states. *Laser Phys Lett* (2024) 21:055209. doi:10.1088/1612-202x/ad3a54

9. Buluta I, Nori F. Quantum simulators. *Science* (2009) 326:108–11. doi:10.1126/science.1177838

10. Song Y, Wu Y, Wu S, Li D, Wen Q, Qin S A quantum federated learning framework for classical clients. *Sci China Phys Mech and Astron* (2024) 67:250311. doi:10.1007/s11433-023-2337-2

11. Song Y, Li J, Wu Y, Qin S, Wen Q, Gao F. A resource-efficient quantum convolutional neural network. *Front Phys* (2024) 12:1362690. doi:10.3389/fphy.2024.1362690

12. Li J, Gao F, Lin S, Guo M, Li Y, Liu H, et al. Quantum k-fold cross-validation for nearest neighbor classification algorithm. *Physica A: Stat Mech its Appl* (2023) 611:128435. doi:10.1016/j.physa.2022.128435

13. Li L, Li J, Song Y, Qin S, Wen Q, Gao F. An efficient quantum proactive incremental learning algorithm. *Sci China Phys Mech and Astron* (2025) 68:210313–9. doi:10.1007/s11433-024-2501-4

14. Childs AM, Kothari R, Somma RD. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J Comput* (2017) 46:1920–50. doi:10.1137/16m1087072

15. Wan L-C, Yu C-H, Pan S-J, Gao F, Wen Q-Y, Qin S-J. Asymptotic quantum algorithm for the toeplitz systems. *Phys Rev A* (2018) 97:062322. doi:10.1103/physreva.97.062322

16. Wan L-C, Yu C-H, Pan S-J, Qin S-J, Gao F, Wen Q-Y. Block-encoding-based quantum algorithm for linear systems with displacement structures. *Phys Rev A* (2021) 104:062414. doi:10.1103/physreva.104.062414

17. Kaplan M, Leurent G, Leverrier A, Naya-Plasencia M. Breaking symmetric cryptosystems using quantum period finding. In: Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference; August 14-18, 2016; Santa Barbara, CA, USA. Proceedings, Part II 36 Springer (2016). p. 207–37.

18. Cai B-B, Wu Y, Dong J, Qin S-J, Gao F, Wen Q-Y. Quantum attacks on 1k-aes and prince. *Computer J* (2023) 66:1102–10. doi:10.1093/comjnl/bxab216

19. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev* (1999) 41:303–32. doi:10.1137/s0036144598347011

20. Bonnetain X, Leurent G, Naya-Plasencia M, Schrottenloher A. Quantum linearization attacks. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security; December 6–10, 2021; Singapore. Proceedings, Part I 27 Springer (2021). p. 422–52.

21. Daemen J (1999). Aes proposal: rijndael

22. Sun X, Tian G, Yang S, Yuan P, Zhang S. Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2023) 42:3301–14. doi:10.1109/tcad.2023.3244885

23. Grassl M, Langenberg B, Roetteler M, Steinwandt R. Applying grover's algorithm to aes: quantum resource estimates. In: *International workshop on post-quantum cryptography*. Springer (2016). p. 29–43.

24. Zou J, Wei Z, Sun S, Liu X, Wu W. Quantum circuit implementations of aes with fewer qubits. In: *Advances in cryptology–ASIACRYPT 2020: 26th international conference on the theory and application of cryptology and information security, daejeon, South Korea, december 7–11, 2020, proceedings, Part II 26*. Springer (2020). p. 697–726.

25. Li Z, Cai B, Sun H, Liu H, Wan L, Qin S, et al. Novel quantum circuit implementation of advanced encryption standard with low costs. *Sci China Phys Mech and Astron* (2022) 65:290311. doi:10.1007/s11433-022-1921-y

26. Jaques S, Naehrig M, Roetteler M, Virdia F. Implementing grover oracles for quantum key search on aes and lowmc. In: Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques; May 10–14, 2020; Zagreb, Croatia. Proceedings, Part II 30 Springer (2020). p. 280–310.

27. Huang Z, Sun S. Synthesizing quantum circuits of aes with lower t-depth and less qubits. In: *International conference on the theory and application of cryptology and information security*. Springer (2022). p. 614–44.

28. Fowler AG. Time-optimal quantum computation (2012). arXiv preprint arXiv:1210.4626.

29. Amy M, Maslov D, Mosca M, Roetteler M. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2013) 32:818–30. doi:10.1109/tcad.2013.2244643

30. Amy M, Maslov D, Mosca M. Polynomial-time t-depth optimization of clifford+ t circuits via matroid partitioning. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2014) 33:1476–89. doi:10.1109/tcad.2014.2341953

31. Chung D, Lee S, Choi D, Lee J. Alternative tower field construction for quantum implementation of the aes s-box. *IEEE Trans Comput* (2021) 71:2553–64. doi:10.1109/tc.2021.3135759

32. Wang Z-G, Wei S-J, Long G-L. A quantum circuit design of aes requiring fewer quantum qubits and gate operations. *Front Phys* (2022) 17:41501. doi:10.1007/s11467-021-1141-2

33. Boyar J, Peralta R. A new combinational logic minimization technique with applications to cryptology. In: Experimental Algorithms: 9th International Symposium, SEA 2010; May 20-22, 2010; Ischia Island, Naples, Italy. Proceedings 9 Springer (2010). p. 178–89.

34. Li Z, Gao F, Qin S, Wen Q. New record in the number of qubits for a quantum implementation of aes. *Front Phys* (2023) 11:1171753. doi:10.3389/fphy.2023.1171753

35. Jang K, Baksi A, Kim H, Seo H, Chattopadhyay A. Improved quantum analysis of speck and lowmc (full version). *Cryptology ePrint Archive* (2022). doi:10.1007/978-3-031-22912-1_23

36. Boyar J, Peralta R. A small depth-16 circuit for the aes s-box. In: *IFIP international information security conference*. Cambridge: Springer (2012). p. 287–98.

37. Liu Q, Preneel B, Zhao Z, Wang M. Improved quantum circuits for aes: reducing the depth and the number of qubits. In: *International conference on the theory and application of cryptology and information security*. Springer (2023). p. 67–98.

38. Nielsen MA, Chuang IL. *Quantum computation and quantum information*. Cambridge University Press (2010).

39. Cong J, Ding Y. Combinational logic synthesis for lut based field programmable gate arrays. *ACM Trans Des Automation Electron Syst (Todaes)* (1996) 1:145–204. doi:10.1145/233539.233540

40. Markov K, Patel I, Hayes J. Optimal synthesis of linear reversible circuits. *Quan Inf Comput* (2008) 8:0282–94. doi:10.26421/qic8.3-4-4

41. Xiang Z, Zeng X, Lin D, Bao Z, Zhang S. Optimizing implementations of linear layers. *IACR Trans Symmetric Cryptology* (2020) 120–45. doi:10.46586/tosc.v2020.i2.120-145

42. Jang K, Baksi A, Kim H, Song G, Seo H, Chattopadhyay A. *Quantum analysis of aes*. Cryptology ePrint Archive (2022). doi:10.62056/ay11zo-3y

43. Shi H, Feng X. Quantum circuits of aes with a low-depth linear layer and a new structure. In: *International conference on the theory and application of cryptology and information security*. Springer (2024). p. 358–95.

44. Selinger P. Quantum circuits of t-depth one. *Phys Rev A—Atomic, Mol Opt Phys* (2013) 87:042302. doi:10.1103/physreva.87.042302