

Article

Optimized Quantum Circuit for Quantum Security Strength Analysis of Argon2

Gyeongju Song , Siwoo Eum, Hyeokdong Kwon, Minjoo Sim , Minwoo Lee and Hwajeong Seo 

Information Computer Engineering, Hansung University, Seoul 02876, Republic of Korea; thdrudwn98@hansung.ac.kr (G.S.); smile267@hansung.ac.kr (S.E.); hyeok@hansung.ac.kr (H.K.); alswnvla@hansung.ac.kr (M.S.); 1771397@hansung.ac.kr (M.L.)

* Correspondence: hwajeong@hansung.ac.kr; Tel.: +82-760-8033

Abstract: This paper explores the optimization of quantum circuits for Argon2, a memory-hard function used in password hashing applications. With the rise of quantum computers, the security of classical cryptographic systems is at risk. This paper emphasizes the need to accurately measure the quantum security strength of cryptographic schemes through highly optimized quantum circuits for the target cryptography algorithm. The proposed method focuses on two perspectives: qubit reduction (qubit-optimized quantum circuit) and depth reduction (depth-optimized quantum circuit). The qubit-optimized quantum circuit was designed to find a point where an appropriate inverse is possible and reuses the qubit through the inverse to minimize the number of qubits. The start and end points of the inverse are determined by identifying a point where qubits can be reused with minimal computation. The depth-optimized quantum circuit reduces the depth of the quantum circuit by using the minimum number of qubits necessary without performing an inverse operation. The trade-off between qubit and depth is confirmed by modifying the internal structure of the circuits and the quantum adders. The qubit optimization achieved up to a 12,229 qubit reduction, while the depth optimization resulted in an approximately 196,741 (approximately 69.02%) depth reduction. In conclusion, this research demonstrates the importance of implementing and analyzing quantum circuits from multiple optimization perspectives. The results contribute to the post-quantum strength analysis of Argon2 and provide valuable insights for future research on optimized quantum circuit design, considering the appropriate trade-offs of quantum resources in response to advancements in quantum computing technology.



Citation: Song, G.; Eum, S.; Kwon, H.; Sim, M.; Lee, M.; Seo, H. Optimized Quantum Circuit for Quantum Security Strength Analysis of Argon2. *Electronics* **2023**, *12*, 4485. <https://doi.org/10.3390/electronics12214485>

Academic Editor: Jian-Qiang You

Received: 12 September 2023

Revised: 27 October 2023

Accepted: 30 October 2023

Published: 31 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: quantum implementation; quantum computing; quantum circuit optimization; Argon2

1. Introduction

Quantum computers have gained attention for their ability to solve specific problems faster than classical computers due to the properties of qubits. The emergence of large-scale quantum computers is anticipated to pose a threat to existing cryptographic systems. Constructing a scalable and fault-tolerant quantum computer is exceptionally challenging. So it may not be an immediate problem, but we need to be prepared to respond to it [1]. In 1994, Peter Shor proposed an algorithm [2] capable of efficiently solving fundamental problems in public-key cryptography, such as integer factorization and discrete logarithms, thereby compromising the security of public-key cryptography. Consequently, the security of target public-key cryptography is no longer guaranteed when large-scale quantum computers capable of performing specific cryptographic attacks appear. In 1996, Lov Grover introduced an algorithm [3]. This algorithm can accelerate brute-force attacks and pre-image attacks on symmetric-key cryptography and hash functions. As a result, it achieves a computational complexity of $O(\sqrt{2^n})$ for finding specific data in unsorted n -bit data. To counter this, the length of the encryption key (hash output length) can be doubled to maintain resistance. However, classical computers and quantum computers differ in

their operation, required resources, and feasible computations, making the security strength of classical computers not directly correspond to the quantum security strength of quantum computers. Accurately measuring the quantum security strength in the context of quantum computers requires optimizing the necessary operations of the specific cryptographic scheme using quantum circuits and accurately verifying the utilized quantum gates and circuit depth. In previous studies, ciphers were implemented as quantum circuits, and required quantum resources were estimated [4–20].

Quantum circuit optimization can be approached from two angles: reducing the number of qubits and minimizing circuit depth. In each implementation, qubit count and depth are often inversely related. Although the number of physically implemented qubits is a key factor in the operation of quantum circuits, in the Noisy Intermediate-Scale Quantum (NISQ) era, it is paramount to reduce depth to mitigate errors, thereby ensuring accurate quantum computing outcomes. An increase in circuit depth not only prolongs computation time but also exacerbates the error rate of qubits.

With this research motivation, this paper proposes two perspectives of optimized quantum circuits for Argon2 and presents estimations of the required quantum resources. These perspectives primarily aim at reducing the number of qubits and minimizing the circuit depth. Each quantum circuit is categorized based on its focus: qubit optimization or depth optimization for specific operations. To delve deeper, we adjust the internal addition mechanism to explore the trade-off between qubit count and circuit depth, striving to pinpoint the most efficient quantum circuit in both domains. In our evaluation, we verify and analyze the resource estimates for both the qubit-optimized and depth-optimized quantum circuits. In a qubit-optimized implementation, we find and set points where inverse operations are possible, and continue to use reusable qubits. In the depth optimization implementation, the minimum number of qubits required for computation is allocated and used without including inverse computation. In addition, an attempt was made to further reduce the depth by changing the adder structure to parallel operation. As a result of optimizing the quantum circuit from both perspectives, the qubit-optimized quantum circuit was reduced by up to 12,229 qubits, and the depth-optimized quantum circuit showed a maximum 196,741 (approximately 69.02%) depth reduction.

The structure of this paper is as follows. In Section 2, related research on quantum computers, the Grover algorithm, and Argon2 is presented to help understand the paper, and Section 3 describes the implementation of the proposed Argon2 quantum circuit. Section 4 estimates and analyzes the resources required for the proposed quantum circuit. Finally, Section 5 concludes the paper with a conclusion.

Our Contribution

In this paper, we implement the core operations used in Argon2 using quantum circuits and estimate quantum resources using the ProjectQ tool [21]. We present four distinct quantum circuits, derived by applying two separate optimization methods—qubit optimization and depth optimization—and by incorporating two different types of adders within each circuit. We demonstrated the qubit–depth trade-offs for the quantum circuits through these two optimized quantum circuits (qubit Opt., depth Opt.). Consequently, we achieved a reduction in the number of qubits by as many as 12,740 and a depth decrease of approximately 89.59%. To the best of our knowledge, this is the first quantum circuit for Argon2. In conclusion, this research emphasizes the significance of analyzing quantum circuits from various optimization perspectives, contributing to post-quantum strength analysis of Argon2 and offering key insights for future quantum circuit design.

2. Background

2.1. Quantum Computer

Quantum computers process data using quantum mechanical phenomena of qubits. These quantum computers can express and process 2^n data at once with n qubits due to the superposition and entanglement properties of qubits, enabling faster calculations

than classic computers. Qubits are controlled through quantum gates, and because of the reversible nature of quantum gates, inverse operations are possible. The following shows H, X, CNOT, and Toffoli matrices among representative quantum gates that control qubits:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad X = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The quantum gate operation of each gate is shown in Figure 1.

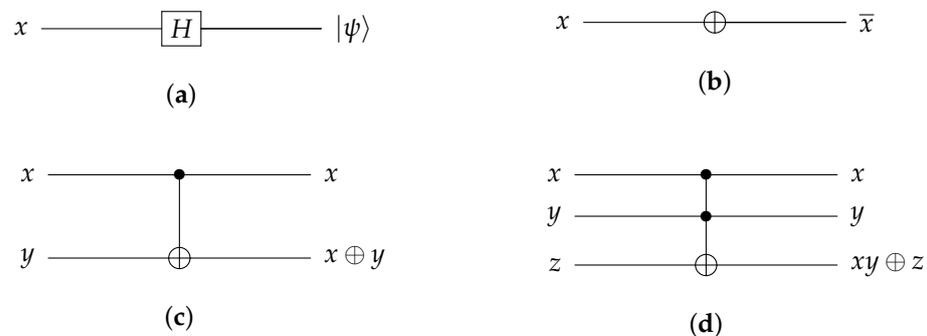


Figure 1. Quantum gates. (a) H gate; (b) X gate; (c) CNOT gate; (d) Toffoli gate.

- (a) H gate: The H gate works with a single qubit and makes the input a superposition.
- (b) X gate: The X gate works with a single qubit and reverses the input.
- (c) CNOT gate: The CNOT gate works with two qubits: control qubit and target qubit. The state of the target qubit y is reversed when the control qubit x is one.
- (d) Toffoli gate: The Toffoli gate works with three qubits: two control qubits and one target qubit. The state of the target qubit z is reversed when the control qubits x and y are both one.

2.2. Grover Algorithm

Quantum computers have the potential to significantly improve the efficiency of certain computational tasks compared to classical computers. One such task is searching for specific n -bit data in an unsorted list, where classical algorithms typically require $O(2^n)$ operations. However, by employing the Grover algorithm on a quantum computer, the search complexity can be reduced to $O(\sqrt{2^n})$. The Grover algorithm for pre-image attacks consists of two main components: an Oracle and a Diffusion operator, shown in Figure 2. This is designed for known-plaintext attacks (KPA) in block ciphers (hash functions), where both the plaintext–ciphertext pairs are known. The Oracle function includes both the hash function $f(x) = y$ and its inverse operation $f^{-1}(x) = y$. When the result of $f(x)$ matches the target hash value y , the Oracle sets $z = 1$. Therefore, when the correct solution is input through this process, the phase of the input state is inverted. The Diffusion operator $U_s = 2|s\rangle\langle s| - I$ (s : the uniform superposition; $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$) is then applied to enhance the probability of observing this state. Through approximately $\lfloor \frac{\pi}{4} \sqrt{N} \rfloor$ (N : search space)

iterations of the Grover algorithm, the probability of measuring the correct solution qubit can be significantly increased.

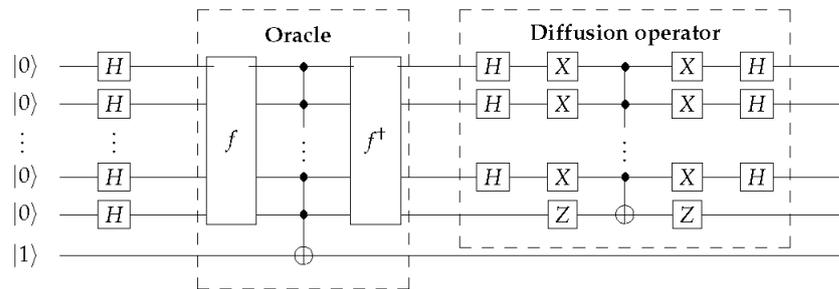


Figure 2. Grover algorithm with $f : \{0,1\}^n \rightarrow \{0,1\}^n$.

2.3. Quantum Adder

In quantum circuits, arithmetic operations necessitate implementation through quantum configurations. Research on the implementation of addition within quantum circuits is extensive, with variations arising based on the quantum gates employed, the number of qubits, and the depth of the circuit. In this paper, we utilize quantum adders proposed in [22]. Specifically, we employ the adder with a depth of $6n - 2$ (referred to as ‘ripple’ in this paper) and the adder with a depth of $2n + 3$ (referred to as ‘simple’ in this paper). For the addition of n -bit inputs $a = a_0, \dots, a_n$, and $b = b_0, \dots, b_n$, the adder stores the addition result in one of the two inputs a and b . In adder, an additional qubit is used to store the carry qubit arising from the previous bit addition. Consequently, the input to the quantum circuit comprises $2n$ qubits corresponding to the two n -bit inputs, along with an ancilla 1-qubit for the temporary storage of the carry. After the one addition, an ancilla carry qubit c is reset to 0, enabling its reuse in subsequent addition operations. After completing the addition, the state of the qubit is $ADD(a, b, c) = (a, a + b, c)$. The two adders have different structures, leading to different depths.

In this paper, the ripple adder was employed for qubit optimization, whereas the simple adder was utilized for depth optimization. As a result, in the ripple adder, a single carry qubit, c , was allocated and reused for all n -bit operations, sequentially performing calculations one bit at a time for the addition of two n -bit inputs. Consequently, there is an increase in depth. In the simple adder, however, n -carry qubits, c , were allocated, and calculations for the addition of two n -bit inputs occurred concurrently across all bits. The carry qubit, c , used in this process is then reused in subsequent additions. As a result, while there is a slight increase in the number of qubits, the depth is reduced.

2.4. Argon2: A Memory-Hard Function for Password Hashing and Other Applications

Argon2 is a key derivation function that won the 2015 Password Hashing Competition. Argon2, a memory-hard function for password hashing and other applications, can be used to hash for credential storage, key derivation, or other applications. It has a simple design that targets fast fill rates of memory and effective use of multiple computing devices while providing defense against trade-off attacks. Argon2 offers three variants: Argon2d, Argon2i, and Argon2id; each variant has the following characteristics:

1. Argon2d: Argon2d uses fast, data-dependent memory accesses, making it highly resistant to GPU cracking attacks and suitable for applications where side-channel timing attacks are not the threat.
2. Argon2i: Argon2i uses data-independent memory access, but is slower, as it uses more memory to protect against trade-off attacks (suitable for password hashing and cipher-based key derivation).
3. Argon2id: Argon2id is a hybrid of Argon2i and Argon2d, using a combination of data-dependent and data-independent memory accesses, giving Argon2i some resistance to

side-channel cache timing attacks and most of Argon2d’s resistance to GPU cracking attacks.

Figure 3 shows the operation of Argon2. Argon2 has two types of inputs: Primary inputs and Secondary inputs or parameters. The Primary inputs are message P and nonce S ; the Secondary inputs are Degree of parallelism p (integer value from 1 to $2^{24} - 1$), Tag length τ (integer number of bytes from 4 to $2^{32} - 1$), Memory size m (integer number of kilobytes from $8p$ to $2^{32} - 1$), Number of iterations t (integer number from 1 to $2^{32} - 1$), Version number v (one byte 0x13), Secret value K (length from 0 to $2^{32} - 1$ bytes), Associated data X (length from 0 to $2^{32} - 1$ bytes), Type y of Argon2 (Argon2d: 0, Argon2i: 1, Argon2id: 2). These are used as inputs to hash function H (in this case Blake2b) and output the result. For parallelism and efficiency, it is divided into several slices which reference previous blocks or join them together, and feed them into functions such as G .

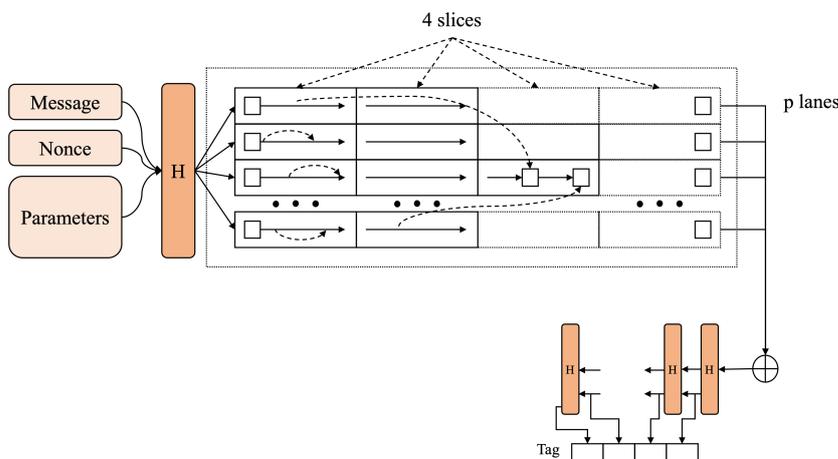


Figure 3. Operation process of Argon2.

2.5. Compression Function G

The compression function G included in H is based on the round function P of Blake2b [23]. The operation of the compression function G is shown in Figure 4. P operates on eight 16-byte registers (128-bit) inputs. The compression function $G(X, Y)$ works with two 1024-byte blocks X and Y . After first calculating $R = X \oplus Y$, R is defined as 16-byte registers R_0 to R_{63} . Then, it is applied to P in row-wise and column-wise order to obtain Z . The P is based on the round function of Blake2b and operates as follows. The eight inputs of 16 bytes each, S_0, S_1, \dots, S_7 , are written as a 4×4 matrix of 64-bit words, where $S_i = (v_{2i+1} || v_{2i})$.

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix}$$

$$\begin{matrix} G(v_0, v_4, v_8, v_{12}) & G(v_1, v_5, v_9, v_{13}) \\ G(v_2, v_6, v_{10}, v_{14}) & G(v_3, v_7, v_{11}, v_{15}) \\ G(v_0, v_5, v_{10}, v_{15}) & G(v_1, v_6, v_{11}, v_{12}) \\ G(v_2, v_7, v_8, v_{13}) & G(v_3, v_4, v_9, v_{14}) \end{matrix}$$

The operation of $G(a, b, c, d)$ is as follows:

$$\begin{aligned}
 a &\leftarrow a + b + 2 \times a_L \times b_L; d \leftarrow (d \oplus a) \ggg 32 \\
 c &\leftarrow c + d + 2 \times c_L \times d_L \\
 b &\leftarrow (b \oplus c) \ggg 24 \\
 a &\leftarrow a + b + 2 \times a_L \times b_L \\
 d &\leftarrow (d \oplus a) \ggg 16 \\
 c &\leftarrow c + d + 2 \times c_L \times d_L \\
 b &\leftarrow (b \oplus c) \ggg 63
 \end{aligned}$$

Finally, G outputs the result of $Z \oplus R$.

$$G : (X, Y) \rightarrow R = X \oplus Y \rightarrow Q \rightarrow Z \rightarrow Z \oplus R$$

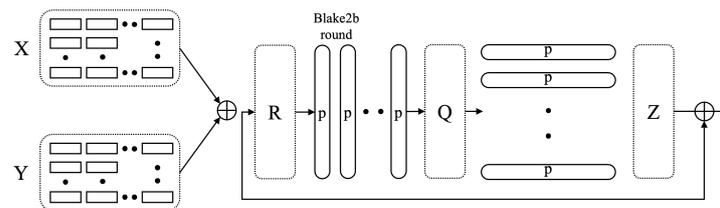


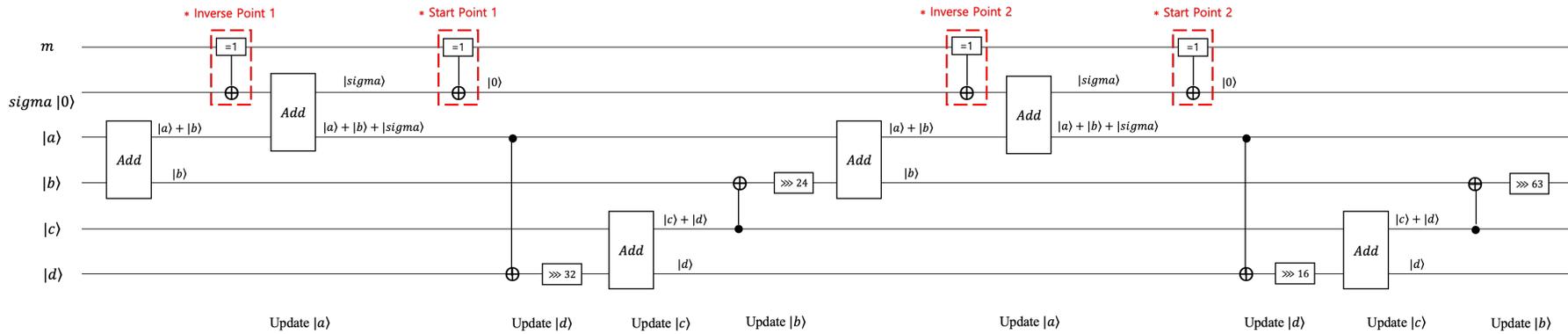
Figure 4. Operation process of compression function G .

3. Optimized Quantum Circuit of Argon2

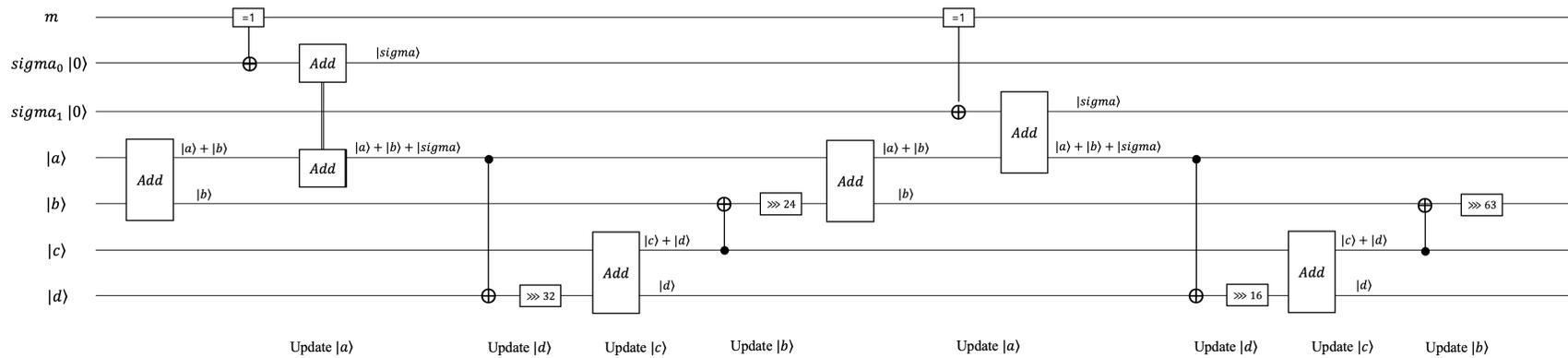
This paper proposes quantum circuits designed for Argon2 and provides estimates for the quantum resources necessary for their operation. This section offers an in-depth discussion of these proposed circuits. For Argon2 quantum circuit implementation, we adopted two primary approaches: qubit reduction (referred to as ‘qubit Opt.’) and depth reduction (referred to as ‘depth Opt.’). The qubit Opt. circuit strategy emphasizes reusing qubits via inverse operations. In contrast, the depth Opt. circuit augments the qubit count, without resorting to inverse operations, to preserve a consistent computational depth. Additionally, this depth-optimized circuit employs a parallel addition design using ancilla qubits to further reduce depth. To optimize this task, we strategically positioned the inverse point. The operating point and starting point of the inverse are set by finding a point where qubits can be reused with minimal computation.

Regarding the quantum adders used within the compression function, we apply both the $(6n - 2)$ -depth adder (referred to as ripple in this paper) and the $(2n + 3)$ -depth adder (referred to as simple in this paper) proposed in [22]. Each adder is applied to both qubit- and depth-optimized quantum circuits, allowing for the examination of the qubit–depth trade-offs. In both perspectives, we adopt a common operation called Classic to Quantum, where the X -gate is applied to quantum data based on the positions where the corresponding classic data have an index of 1. This operation is employed to reduce the number of ancilla qubits. To reduce quantum circuit depth, the Shift operation is performed by changing the indices of the array rather than using Swap gates. Figure 5 shows optimized quantum circuits for G : (1) qubit-optimized quantum circuit and (2) depth-optimized quantum circuit. For the two circuits in the figure, input m is pre-determined classic data, σ is pre-determined quantum data, and $|a\rangle$ to $|d\rangle$ are the quantum data input to G . Detailed explanations of the two quantum circuits are provided in Sections 3.1 and 3.2. The order in which $|a\rangle$ to $|d\rangle$ are input to G is as follows (The inputs to function G , denoted as v_0 to v_{15} , represent 64 qubits. These qubits are updated within this function.):

1. $G(a, b, c, d) = G(v_0, v_4, v_8, v_{12})$;
2. $G(a, b, c, d) = G(v_1, v_5, v_9, v_{13})$
3. $G(a, b, c, d) = G(v_2, v_6, v_{10}, v_{14})$;
4. $G(a, b, c, d) = G(v_3, v_7, v_{11}, v_{15})$
5. $G(a, b, c, d) = G(v_0, v_5, v_{10}, v_{15})$;
6. $G(a, b, c, d) = G(v_1, v_6, v_{11}, v_{12})$
7. $G(a, b, c, d) = G(v_2, v_7, v_8, v_{13})$;
8. $G(a, b, c, d) = G(v_3, v_4, v_9, v_{14})$



(1) Qubit-optimized Quantum circuit for Q



(2) Depth-optimized Quantum circuit for Q

Figure 5. Optimized quantum circuits for G: (1) qubit-optimized quantum circuit; (2) depth-optimized quantum circuit; * symbol indicates the inverse and start points.

3.1. Qubit-Optimized Quantum Circuit

The qubit-optimized quantum circuit reuses qubits through reverse operation, increasing the depth at the cost of reducing the number of qubits. This method reuses the used 64-qubit *sigma* through inverse operation so that all compression functions operate as a single 64-qubit *sigma*. The quantum circuit for this can be seen in (1) of Figure 5. In this circuit, there are two reverse points to reduce the number of qubits, and the *sigma* is reset to $|0\rangle$ at both points. Allocated qubits for *sigma* are not only reused in functions but are still available in all rounds. Including the inverse operation, the quantum data $|a\rangle$ to $|d\rangle$ are updated according to the order.

Algorithm 1 shows the operation of the qubit-optimized quantum circuit for the compression function G. Lines 3 and 5 and lines 18 and 21 represent the reverse start and end points, and lines 6 and 23 indicate the timing of the reverse operation of each reverse point. In lines 1, 4, 12, 18, 21, and 27, ADD is implemented using two adders: depth $6n - 2$ and depth $2n + 3$, and the difference between each adder is shown in Section 4. The depth was not increased by adjusting the operation index order instead of shift, and the depth was reduced by adjusting the physical location of qubits using a logical array instead of a SWAP gate. The Classic to Quantum function in lines 4 and 20 is designed so that m and *sigma* are not quantum-to-quantum operations between qubits, but classic-to-quantum operations according to the state of classic constant values. This approach reduces the number of qubits and quantum gates used for m and *sigma* updates. Since constant m is a known constant, m is stored in the pre-computation table, and the X gate is operated at the same *sigma* index as the part where the index bit value of m is one in each round. These operations are also very efficient in terms of quantum resources, as they can be replaced with the use of a low-cost X gate rather than the CNOT gate.

Algorithm 1 Qubit-optimized quantum circuit for the compression function (G).

```

Input:  $a, b, c, d, sigma$ 
 $a \leftarrow \text{ADD}(b, a)$ 
*Reverse Start Point1
   $sigma \leftarrow \text{Classic to Quantum}(m[sigma[r][2 * i + 0]])$ 
 $a \leftarrow \text{ADD}(sigma, a)$ 
*Reverse End Point1
Start reverse1(Start to End)
for (k=0 to length(d)) :
   $d[k] \leftarrow \text{CNOT}(a[k], d[k])$ 
for (k=0 to 64) :
   $d_{box1}.append(d[(k + 32) \bmod 64])$ 
 $d = d_{box1}$ 
 $c \leftarrow \text{ADD}(d, c)$ 
for (k=0 to length(b)) :
   $b[k] \leftarrow \text{CNOT}(c[k], b[k])$ 
for (k=0 to 64) :
   $b_{box1}.append(b[(k + 24) \bmod 64])$ 
 $b = b_{box1}$ 
 $a \leftarrow \text{ADD}(b, a)$ 
*Reverse Start Point2
   $sigma \leftarrow \text{Classic to Quantum}(m[sigma[r][2 * i + 1]])$ 
 $a \leftarrow \text{ADD}(sigma, a)$ 
*Reverse End Point2
Start reverse2(Start to End)
for (k=0 to 64) :
   $d_{box2}.append(d[(k + 16) \bmod 64])$ 
 $d = d_{box2}$ 

```

Algorithm 1 Cont.

```

c ← ADD(d, c)
for (k=0 to 64) :
    bbox2.append(b[(k + 64) mod 64])
b = bbox2

```

3.2. Depth-Optimized Quantum Circuit

Depth-optimized quantum circuits increase the use of ancilla qubits but decrease the depth. The 64-qubit *sigma* used is not reused; it is allocated and used whenever *sigma* is used in any function. The quantum circuit for this can be seen in Figure 5(2). Since there are no reverse actions, there is no reverse point. The qubits assigned to *sigma* are non-reusable, so it continues to be assigned in all rounds, not just in the function. Without including the inverse operation, the quantum data $|a\rangle$ to $|d\rangle$ are updated according to the order.

Algorithm 2 shows the pseudo-code for a depth-optimized quantum circuit for compression function *G*. In lines 2, 4, 10, 16, 18, and 22, ADD is implemented using two adders: depth $6n - 2$ adder and depth $2n + 3$ adder, and the difference between each adder is shown in Section 4. The depth was not increased by adjusting the operation index order instead of shift, and the depth was reduced by adjusting the physical location of qubits using a logical array instead of a SWAP gate. The Classic to Quantum function in lines 3 and 17 is designed so that *m* and Sigma are not quantum-to-quantum operations between qubits, but classic-to-quantum operations according to the state of classic constant values. This method does not involve an inverse operation, allowing the total depth to be reduced. As with qubit-optimized quantum circuits, the known constant *m* is stored in the pre-computation table, and the X gate is operated at the same sigma index as the part where the index bit value of *m* is one in each round. These operations are also very efficient in terms of quantum resources, as they can be replaced with the use of a low-cost X gate rather than the CNOT gate.

Algorithm 2 Depth-optimized quantum circuit for the compression function (*G*).

```

Input: a, b, c, d, sigma1, sigma2
a ← ADD(b, a)
sigma1 ← Classic_to_Quantum(m[sigma1[r][2 * i + 0]])
a ← ADD(sigma1, a)
for (k=0 to length(d)) :
    d[k] ← CNOT(a[k], d[k])
for (k=0 to 64) :
    dbox1.append(d[(k + 32) mod 64])
d = dbox1
c ← ADD(d, c)
for (k=0 to length(b)) :
    b[k] ← CNOT(c[k], b[k])
for (k=0 to 64) :
    bbox1.append(b[(k + 24) mod 64])
b = bbox1
a ← ADD(b, a)
sigma2 ← Classic_to_Quantum(m[sigma2[r][2 * i + 1]])
a ← ADD(sigma2, a)
for (k=0 to 64) :
    dbox2.append(d[(k + 16) mod 64])
d = dbox2
c ← ADD(d, c)
for (k=0 to 64) :
    bbox2.append(b[(k + 64) mod 64])
b = bbox2

```

4. Evaluation

This paper proposes two perspectives on the design of quantum circuits for Argon2, focusing on qubit reduction (referred to as ‘qubit optimization’) and depth reduction (referred to as ‘depth optimization’). The qubit-optimized quantum circuit minimizes the number of qubits by reusing them through inverse operations, while the depth-optimized quantum circuit maintains the same level of computational depth by increasing the number of qubits without using inverse operations. Additionally, the internal structure of the quantum circuits is modified to explore the optimal design for both qubit number and depth, analyzing the trade-off between the two. Four distinct quantum circuits are presented, each representing one of the two optimization perspectives and one of the two variations of adder circuits. The quantum resources required for these quantum circuits were estimated using the ProjectQ tool [21].

The estimated results are presented in Tables 1–5. Table 1 provides the estimated quantum resources for each optimized function in Argon2. ‘Qubit Opt’ and ‘Depth Opt’ represent the qubit-optimized and depth-optimized quantum circuits, and ‘ripple’ and ‘simple’ are the $(2n + 3)$ depth and $(6n - 2)$ depth adders proposed in [22]. The 1qClifford indicates 1 qubit Clifford gate; here it represents the X gate. The results indicate that the qubit-optimized G operates with 1089 qubits, and the depth-optimized G operates with 13,318 qubits, demonstrating a reduction of up to 12,229 qubits through qubit optimization. The depth per round for the qubit-optimized circuit is 74,713 and 220,033, depending on the adder used, while the depth per round for the depth-optimized G is 23,401 (approximately a 68.68% reduction) and 68,163 (approximately a 69.02% reduction), depending on the adder. This suggests a potential reduction of up to approximately 69.02% in depth through depth optimization. Tables 2–5 present the estimated quantum resources for each step of Argon2. Among the steps, blake2b utilizes the most resources.

In summary (shown in Table 6), selecting the qubit-optimized quantum circuit can reduce the number of qubits by up to 12,740, with the flexibility to choose the adder based on optimization needs. Choosing the depth-optimized quantum circuit can reduce the depth by up to approximately 89.59%, with the possibility of selecting the adder based on the optimization perspective. The results of this paper indicate that selecting the ripple adder in the depth-optimized quantum circuit minimizes the depth. This paper has confirmed through our results that the quantum resources used can be determined according to the optimization perspective of quantum circuit implementation by modifying the operation sequence and structure. We have made efforts to design the operation sequence and structure of the quantum circuit to reduce the number of qubits and depth (qubit optimization and depth optimization), and we have achieved such results.

Table 1. Estimation results of quantum resources for each optimized function in Argon2. The result is a measure of the amount of resources per round (Qubit Opt: qubit-optimized quantum circuit, Depth Opt: depth-optimized quantum circuit).

Operation	Adder	#Qubit	#1qClifford	#CNOT	#Toffoli	Full Depth
G (Qubit Opt)	ripple	1089	70,836	204,864	72,000	74,713
G (Qubit Opt)	simple	1089	22	172,032	72,576	220,033
G (Depth Opt)	ripple	13,318	70,284	204,864	72,000	23,401
G (Depth Opt)	simple	13,318	12	172,032	72,576	68,163
$Z \oplus R$	-	1536	-	1024	-	2

Table 2. Quantum resource estimation results for steps in Argon2 (qubit Opt. adder: ripple).

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	Full Depth
Initial				(None)	
Update	1090			(None)	
Final		1.62×2^{17}	1.17×2^{19}	1.64×2^{17}	1.71×2^{17}
Blake2b		1.51×2^{22}	1.1×2^{24}	1.54×2^{22}	1.6×2^{22}
Total		1.51×2^{22}	1.14×2^{24}	1.59×2^{22}	1.65×2^{22}

Table 3. Quantum resource estimation results for steps in Argon2 (qubit Opt. adder: simple).

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	Full Depth
Initial				(None)	
Update	1090			(None)	
Final		1.03×2^6	1.98×2^{18}	1.66×2^{17}	1.25×2^{19}
Blake2b		1.93×2^{10}	1.85×2^{23}	1.55×2^{22}	1.18×2^{24}
Total		1.99×2^{10}	1.91×2^{23}	1.6×2^{22}	1.21×2^{24}

Table 4. Quantum resource estimation results for steps in Argon2 (depth Opt. adder: ripple).

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	Full Depth
Initial				(None)	
Update	13,830			(None)	
Final		1.6×2^{17}	1.17×2^{19}	1.64×2^{17}	1.42×2^{14}
Blake2b		1.5×2^{22}	1.1×2^{24}	1.54×2^{22}	1×2^{21}
Total		1.55×2^{22}	1.14×2^{24}	1.59×2^{22}	1.01×2^{21}

Table 5. Quantum resource estimation results for steps in Argon2 (depth Opt. adder: simple).

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	Full Depth
Initial				(None)	
Update	13,830			(None)	
Final		1.12×2^5	1.98×2^{18}	1.66×2^{17}	1.56×2^{17}
Blake2b		1.05×2^{10}	1.85×2^{23}	1.55×2^{22}	1.46×2^{22}
Total		1.08×2^{10}	1.91×2^{23}	1.6×2^{22}	1.51×2^{22}

Table 6. Quantum resource estimation results in Argon2.

Operation (Opt., Adder)	#Qubit	#1qClifford	#CNOT	#Toffoli	Full Depth
Argon2 (qubit, ripple)	1090	1.51×2^{22}	1.14×2^{24}	1.59×2^{22}	1.65×2^{22}
Argon2 (qubit, simple)	1090	1.99×2^{10}	1.91×2^{23}	1.6×2^{22}	1.21×2^{24}
Argon2 (depth, ripple)	13,830	1.55×2^{22}	1.14×2^{24}	1.59×2^{22}	1.01×2^{21}
Argon2 (depth, simple)	13,830	1.08×2^{10}	1.91×2^{23}	1.6×2^{22}	1.51×2^{22}

5. Conclusions

This paper presents quantum circuits from two perspectives for Argon2. In the qubit-optimized quantum circuit, the number of qubits is reduced by reusing previously used qubits through inverse operations, but the depth increases due to the computations required for the inverse. Conversely, the depth-optimized quantum circuit augments the number of qubits by employing ancilla qubits and parallel adder structures, avoiding

inverse operations and thus markedly reducing the depth. Our quantum resource analysis indicates a disparity of up to 12,740 qubits and 196,741 in depth between the four variations of qubit-optimized and depth-optimized quantum circuits. Given the current limitations of imperfect fault-tolerant quantum computers, it is necessary to analyze the post-quantum resistance strength through the implementation of quantum circuits from various perspectives. By appropriately adjusting the trade-off between qubits and depth, the most suitable quantum circuit can be identified. Therefore, the implementation and analysis of quantum circuits from various optimization perspectives are crucial research areas. The results of this paper contribute to the post-quantum strength analysis of Argon2 and provide insights for future research on quantum circuit design with appropriate trade-offs of quantum resources in response to advancements in quantum computing technology.

Author Contributions: Software, G.S.; Investigation, H.K. and M.S.; Writing—original draft, G.S.; Writing—review & editing, S.E., M.L. and H.S.; Supervision, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by Hansung University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aumasson, J.P. The impact of quantum computing on cryptography. *Comput. Fraud Secur.* **2017**, *2017*, 8–11. . [CrossRef]
2. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
3. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on THEORY of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
4. Grassl, M.; Langenberg, B.; Roetteler, M.; Steinwandt, R. Applying Grover’s algorithm to AES: Quantum resource estimates. In *Post-Quantum Cryptography*; Springer: Cham, Switzerland, 2016; pp. 29–43.
5. Almazrooie, M.; Samsudin, A.; Abdullah, R.; Mutter, K.N. Quantum reversible circuit of AES-128. *Quantum Inf. Process.* **2018**, *17*, 112. [CrossRef]
6. Anand, R.; Maitra, A.; Mukhopadhyay, S. Grover on SIMON. *Quantum Inf. Process.* **2020**, *19*, 340. [CrossRef]
7. Chauhan, A.K.; Sanadhya, S.K. Quantum resource estimates of grover’s key search on aria. In Proceedings of the Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, 17–21 December 2020; Proceedings 10; Springer: Cham, Switzerland, 2020; pp. 238–258.
8. Luo, Q.; Li, Q.; Li, X.; Yang, G.; Shen, J.; Zheng, M. Quantum implementaion of SM4 block cipher with less qubits. *Res. Sq.* **2023**, preprint. [CrossRef]
9. Baksi, A.; Jang, K.; Song, G.; Seo, H.; Xiang, Z. Quantum implementation and resource estimates for rectangle and knot. *Quantum Inf. Process.* **2021**, *20*, 395. [CrossRef]
10. Anand, R.; Maitra, A.; Mukhopadhyay, S. Evaluation of quantum cryptanalysis on speck. In Proceedings of the Progress in Cryptology–INDOCRYPT 2020: 21st International Conference on Cryptology in India, Bangalore, India, 13–16 December 2020; Proceedings 21; Springer: Cham, Switzerland, 2020; pp. 395–413.
11. Jang, K.; Baksi, A.; Breier, J.; Seo, H.; Chattopadhyay, A. Quantum implementation and analysis of default. *Cryptol. ePrint Arch.* **2022**, 1–17. [CrossRef]
12. Rahman, M.; Paul, G. Grover on KATAN: Quantum resource estimation. *IEEE Trans. Quantum Eng.* **2022**, *3*, 3100809. [CrossRef]
13. Jang, K.; Song, G.; Kim, H.; Kwon, H.; Kim, H.; Seo, H. Parallel quantum addition for Korean block ciphers. *Quantum Inf. Process.* **2022**, *21*, 373. [CrossRef]
14. Huang, Z.; Sun, S. Synthesizing quantum circuits of AES with lower t-depth and less qubits. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 5–9 December 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 614–644.
15. Jang, K.; Baksi, A.; Song, G.; Kim, H.; Seo, H.; Chattopadhyay, A. Quantum Analysis of AES. *Cryptol. ePrint Arch.* **2022**. Available online: <https://eprint.iacr.org/2022/683> (accessed on 31 October 2023).
16. Song, G.; Jang, K.; Kim, H.; Seo, H. A Parallel Quantum Circuit Implementations of LSH Hash Function for Use with Grover’s Algorithm. *Appl. Sci.* **2022**, *12*, 10891. [CrossRef]
17. Song, G.; Jang, K.; Kim, H.; Lee, W.K.; Hu, Z.; Seo, H. Grover on SM3. In Proceedings of the Information Security and Cryptology–ICISC 2021: 24th International Conference, Seoul, Republic of Korea, 1–3 December 2021; Revised Selected Papers; Springer: Cham, Switzerland, 2022; pp. 421–433.
18. Zou, J.; Li, L.; Wei, Z.; Luo, Y.; Liu, Q.; Wu, W. New quantum circuit implementations of SM4 and SM3. *Quantum Inf. Process.* **2022**, *21*, 181. [CrossRef]

19. Song, G.; Jang, K.; Kim, H.; Eum, S.; Sim, M.; Kim, H.; Lee, W.; Seo, H. SPEEDY quantum circuit for Grover's algorithm. *Appl. Sci.* **2022**, *12*, 6870. [[CrossRef](#)]
20. Song, G.; Jang, K.; Seo, H. Improved Low-Depth SHA3 Quantum Circuit for Fault-Tolerant Quantum Computers. *Appl. Sci.* **2023**, *13*, 3558. [[CrossRef](#)]
21. Steiger, D.S.; Häner, T.; Troyer, M. ProjectQ: An open source software framework for quantum computing. *Quantum* **2018**, *2*, 49. [[CrossRef](#)]
22. Cuccaro, S.A.; Draper, T.G.; Kutin, S.A.; Moulton, D.P. A new quantum ripple-carry addition circuit. *arXiv* **2004**, arXiv:quant-ph/0410184.
23. Aumasson, J.P.; Neves, S.; Wilcox-O'Hearn, Z.; Winnerlein, C. BLAKE2: Simpler, smaller, fast as MD5. In Proceedings of the Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, 25–28 June 2013; Proceedings 11; Springer: Berlin/Heidelberg, Germany, 2013; pp. 119–135.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.