Qeios

A Generalized Space-Efficient Algorithm for Quantum Bit String Comparators

Khuram Shahzad and Omar Usman Khan

Dept. of Computer Science, National University of Computer and Emerging Sciences, Peshawar, Pakistan

Quantum Bit String Comparators (QBSC) operate on two sequences of n-qubits, enabling the determination of their relationships, such as equality, greater than, or less than. This is analogous to the way conditional statements are used in programming languages. Consequently, QBSCs play a crucial role in various algorithms that can be executed or adapted for quantum computers. The development of efficient and generalized comparators for any n-qubit length has long posed a challenge, as they have a high-cost footprint and lead to quantum delays.

Comparators that are efficient are associated with inputs of fixed length. As a result, comparators without a generalized circuit cannot be employed at a higher level, though they are well-suited for problems with limited size requirements. In this paper, we introduce a generalized design for the comparison of two n-qubit logic states using just two ancillary bits. The design is examined on the basis of qubit requirements, ancillary bit usage, quantum cost, quantum delay, gate operations, and circuit complexity, and is tested comprehensively on various input lengths. The work allows for sufficient flexibility in the design of quantum algorithms, which can accelerate quantum algorithm development.

1 Introduction

Quantum computing and its related algorithms have witnessed remarkable advances in recent years, owing to the principles of quantum mechanics and enhanced computing power. They offer the potential to efficiently solve many mathematical problems that are intractable for classical computers. For instance, Shor's algorithm[7] can achieve polynomial-time solutions for hard problems such as integer factorization or discrete logarithms. Similarly, Grover's algorithm[7] can provide a quadratic speedup for an unstructured search problem.

The quantum bit string comparator (QBSC) is a crucial component in quantum algorithms as it incorporates conditional statements, expanding the range of applications for quantum algorithms. It allows quantum programmers to utilize successful techniques from the classical computation that rely on comparisons [26]. In a classical sense, a conditional statement leads to two mutually exclusive states. However, in the quantum domain, such conditional statements may lead to the merging of both branches due to superposition. Designing a comparator for quantum circuits is a significant challenge for researchers in the field. Existing quantum comparators are not scalable in terms of input size, as the circuit size depends on the number of inputs. Therefore, these quantum comparators, which lack a generalized circuit, are not suitable for higher-level applications, although they can handle problems with small size requirements. However, many applications, such as integer factorization, optimization, option pricing [11], and risk analysis, often require one of the inputs to be classical.[1] This necessitates a generalization of the circuit, especially when further computations are involved.

Numerous approaches have been put forth to effectively devise quantum comparators. These include the serial-based approach [22, 23, 26, 27], the tree-based approach [24, 25] and the Quantum Fourier transform (QFT) [1]. In the serial-based quantum comparator, comparisons of quantum bits are executed sequentially from the least significant bit to the most significant bit. Conversely, a tree-based quantum comparator can evaluate quantum bits for comparison in

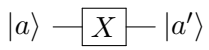

Figure 1: NOT Gate representation for $|a'\rangle = X\,|a\rangle$

parallel. Although it holds an advantage over the serial-based approach in terms of time delay, it still falls short in terms of quantum cost. However, both approaches require one or two bits for comparison, along with two or more ancillary bits. This current study presents a generalized quantum comparator having a minimal quantum cost and optimized quantum delay that can compare two classical numbers of any size in binary form using a quantum circuit. The data on qubits remains unchanged and can also be used for other operations. Our quantum comparator has a linear scaling of Qubit resources with respect to the size of the input numbers. For example, it takes two ancillary qubits to compare two n-bit numbers. This is an improvement over some existing quantum comparators that require more ancillary qubits or have a higher quantum cost and higher quantum delays [22, 23, 24, 25, 26, 27]. The research analyzes the comparator qubit requirements, gate operations, and circuit complexity for various input sizes. Our proposed comparator's performance is comprehensively analyzed with respect to quantum cost, quantum delay, and ancillary bits. Since quantum comparators are fundamental in many quantum algorithms and applications; e.g. integer factorization, optimization, option pricing, and risk analysis, the work holds good promise for the design and development of quantum algorithms [1].

1.1 Quantum Gates for GQBSC

Quantum gates are analogous to the application of various transformations on input states represented through qubits. In this section, we describe some fundamental gates that are used in GQBSCs. Each of the gates here performs some specific unitary operation, and as such, they are also represented using unitary matrices [28, 29].

A single-qubit unitary case is that of the NOT gate, which is used for bit-flip operations, and is given in Fig. 1 along with its mathematical representation. The gate represents the Pauli X operator and exhibits the properties $X^2 = I$, where I is the Pauli Identity operator. The bit-flip here is geometrically interpreted a half turn about the x-axis in a Bloch sphere.

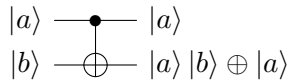

Figure 2: CX Gate representation for $|a\rangle\,|b\rangle = |a\rangle\,|b\rangle \oplus |a\rangle$

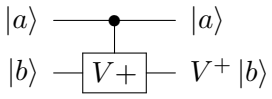

Figure 3: Controlled V^+ Gate representation for $|a\rangle\,|b\rangle \mapsto V^+(|b\rangle)$ (unitary transformation)

Moving on, we have the CX gate, also known as the CNOT gate, which is a two-qubit gate having one control and one target qubit. It performs a NOT operation on the target qubit if the control qubit is in the state $|1\rangle$, and is given in Fig. 2. Here, the black dot within the CNOT gate signifies that when the controlling bit (represented by the black dot) holds the value *1*, the NOT gate on the target bit is activated. Conversely, the white dot indicates that when the controlling bit is *0* the NOT gate operates on the target bit. Both the NOT and the Controlled NOT are fundamental gates for having a quantum cost and delay of 1 [27]. We use the definition of quantum cost as the number of fundamental unitary and binary reversible gates (e.g. NOT and CNOT) that are used in the design of a gate in a general decomposition sense [24]. This is then extended to quantum delay, defined as the measure of the logical depth of a circuit, such that the delay of fundamental unitary and binary reversible gates would be 1Δ [24].

V and V^+ represent two useful quantum gates with a quantum cost and delay each equivalent to 1 [24, 27]. These gates are defined by their respective unitary matrices, which adhere to the mathematical property $VV^+ = V^+V = 1$, signifying their unitarity and the inherent reversibility of quantum operations. Geometrically, these gates are interpreted as a quarter turn about the x-axis in a Bloch sphere. These matrices are pre-

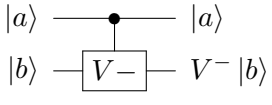

Figure 4: Controlled V Gate representation for $|a\rangle\,|b\rangle \mapsto V^-(|b\rangle)$ (unitary transformation)

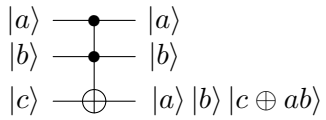

Figure 5: CCX Gate representation for $|a\rangle\,|b\rangle\,|c\rangle = |a\rangle\,|b\rangle\,|c \oplus ab\rangle$

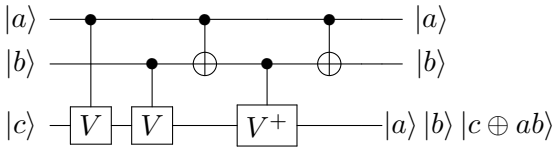

Figure 6: CCX Gate Representation using V and V+ gate

cisely defined as:

$$V = \frac{i+1}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \tag{1}$$

$$V^+ = \frac{1-i}{2}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \tag{2}$$

where i represents the imaginary unit.

A ternary case is that of the CCX gate; also known as the Toffoli gate. It operates on 3-qubits by performing a bit-wise logical AND on the first two qubits and then applies its result to the third qubit as an XOR, resulting in a flip in $|c\rangle$, provided that both the first and second qubits are in state $|1\rangle$. This is illustrated in Fig. 5. The quantum cost of a Toffoli gate is determined by considering the number of fundamental quantum gates it comprises, such as CNOT gates, controlled-V gates, and controlled-V^+ gates. In the case of a Toffoli gate, it involves two CNOT gates, two controlled-V gates, and one controlled-V^+ gate. Therefore, both the quantum cost and delay of a Toffoli gate are evaluated at $5\triangle$ [24, 27]. The CCX can be realized through the use of XOR and AND operations. Considering the case of $|a\rangle$, $|b\rangle$, and $|c\rangle$ (Figure 5), then CCX can be achieved as $|a\rangle\,|b\rangle\,|c \oplus ab\rangle$. This is realized in the first step through an AND $\mathbf{x} = \mathbf{a_2} \wedge \mathbf{b_2}$. The intermediate $\mathbf{x}$ is then passed as an XOR to get the final result $\mathbf{x'} = \mathbf{c} \oplus \mathbf{x}$.

In the next sections, we discuss techniques that implement the GQBSC using these gates.

2 Related Work

There have been various quantum comparators reported in the literature. Wang et al. introduced a quantum comparator that accomplishes the comparison of two quantum logic states, each having n quantum bits, in a sequential manner [22]. The circuit involves the use of $2n-2$ ancillary input bits, as depicted in (Figure 7). In this diagram, $|e_0\rangle$ and $|e_1\rangle$ represent the meaningful outcomes. Al-Rabadi proposed a sequentially structured quantum comparator that links together a sequence of 1-bit comparators, as shown in (Figure 8). This arrangement requires 6 ancillary input bits for each 1-bit comparator [23]. Thapliyal et al. devised a tree-based comparator, illustrated in (Figure 10), wherein every node corresponds to a 2-bit comparator demonstrated in (Figure 10). This specialized comparator can assess 2-bit binary numbers [24]. Vudadha et al. enhanced the tree-based comparator using a prefix tree [25]. This design comprises three stages: the initial stage integrates a 1-bit comparator featuring two meaningful outputs. The outputs of the 1-bit comparator phase are then grouped in the second stage using prefix grouping to generate the final outputs G. While the tree-based quantum comparator surpasses the sequential-based comparator in terms of time delay, it lags behind the sequential-based comparator in the number of ancillary bits required. Xia et al. proposed a new serial base comparator (Figure 9) that can compare two n-size numbers using one ancillary qubit, but the quantum cost and delay are still high [27].

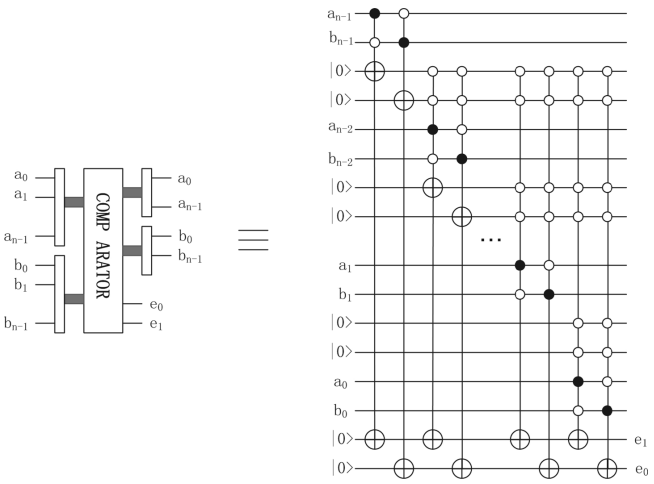

Figure 7: The reversible comparator presented by Want et al. [22].

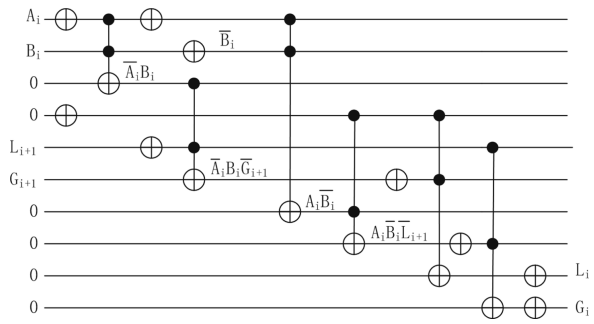

Figure 8: The comparator presented by Al-Rabadi et al. [23].

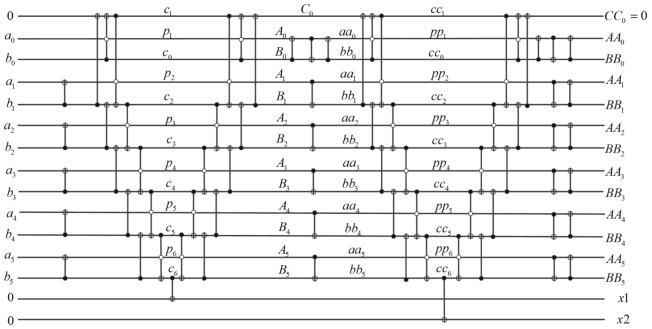

Figure 9: The comparator presented by Haiying Xia et al.[27].

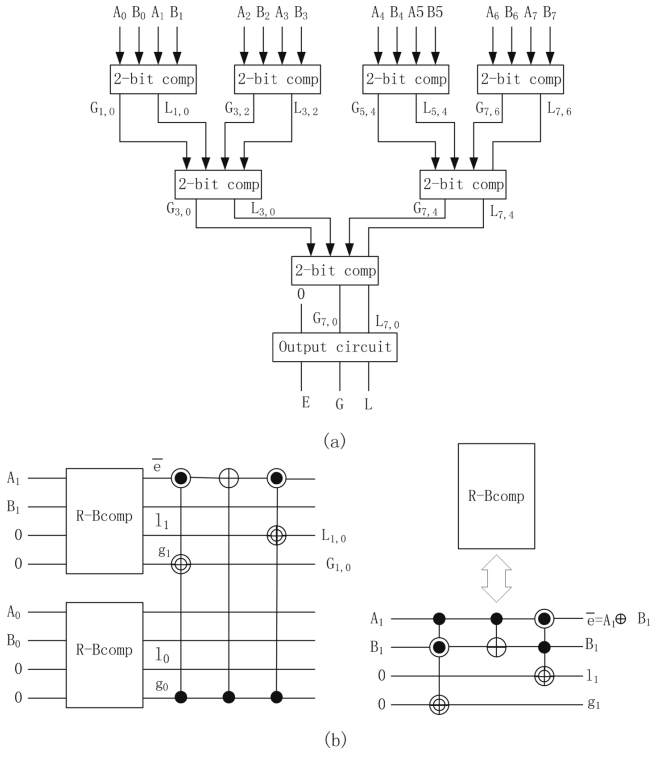

Figure 10: There are (a) and (b) the tree-based comparator presented by Thapliyal et al. [24].

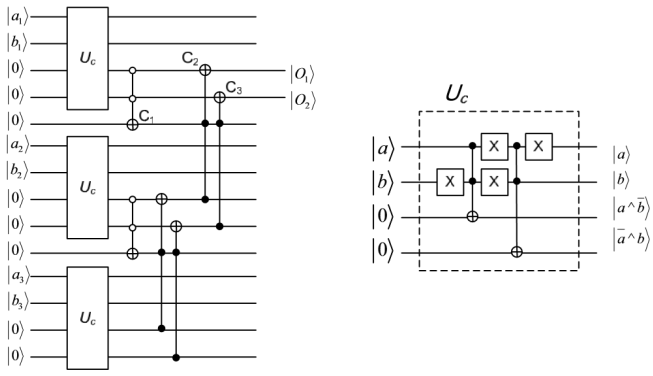

Figure 11: The comparator presented by David et al.[26].

3 Proposed GQBSC

Our objective is to conduct a comparison between two n-bit numbers, denoted as a and b, utilizing only two ancillary bits. We employ two binary strings of length n such that a is represented as $a_{n-1} \dots a_0$, where a_0 is the most significant bit. Similarly, b is represented as $b_{n-1} \dots b_0$, where b_0 is the most significant bit. In case the strings are unequal, padding is performed on the smaller string up to the length of the larger string. To characterize the outputs of a_i, and b_i, we now present a discussion of our proposed approach, described in Algo. 1.

The approach takes as input the vectors $|a\rangle$, $|b\rangle$, and registers $|r_0\rangle$, $|r_1\rangle$ initialized to $|0\rangle$. The algorithm returns $|r_0\rangle$, $|r_1\rangle$ as classical variables c_0 and c_1. The algorithm works by performing a 1-bit quantum bit string comparator (Algo. 2) on the most significant bit a_0 and b_0 and collects its intermediate output in r_0 and r_1. For each of the remaining bit string lengths, the intermediate outputs are approximated against $|0\rangle$, and if true, the 1-bit GQBSC is carried out for the next significant bit. However, if r_1 is approximated to $|1\rangle$, a bit flip is carried out on r_0 (through Pauli X gate).

The 1-bit GQBSC (Algo. 2) is the realization of the quantum circuit given in Fig. 12. It is essentially a core component of our proposed model and utilizes four qubits; $|a\rangle$ and $|b\rangle$ for storing the two input bitstrings, and $|r_0\rangle$ and $|r_1\rangle$ for storing intermediate comparisons. The comparison is itself carried out through the successive application of multiple NOT and CCX gates as shown. The intermediate states are then stored in classical registers cr through Z-measurement gates. The output of the proposed circuit exhibits four

possible binary states; *00* indicating $a \approx b$, *10* indicating $a > b$, and *01* and *11* both indicating $a < b$.

Data: $|a\rangle$, $|b\rangle$, $|r_0\rangle = |r_1\rangle = |0\rangle$
Result: c_0, c_1
$[r_0, r_1] \leftarrow$ 1BITGQBSC(a_0, b_0, r_0, r_1)
for i *in* $1 \ldots n-1$ **do**
  if $(r_0 \approx |0\rangle)$ *and* $(r_1 \approx |0\rangle)$ **then**
    $[r_0, r_1] \leftarrow$ 1BITGQBSC(a_i, b_i, r_0, r_1)
  end
  if $(r_1 \approx |1\rangle)$ **then**
    $r_0 \leftarrow \mathrm{X}r_0$
  end
end
$[c_0, c_1] \leftarrow [r_0, r_1]$

Algorithm 1: Generalized Quantum Bit String Comparator(GQBSC)

Result: r_0, r_1
1BITGQBSC $(|a\rangle, |b\rangle, r_0, r_1)$
  $b \leftarrow \mathrm{X}b$
  $r_0 \leftarrow r_0 \oplus (a_1 \wedge b_1)$
  $a \leftarrow \mathrm{X}a$
  $b \leftarrow \mathrm{X}b$
  $r1 \leftarrow r1 \oplus (a_1 \wedge b_1)$
  $a \leftarrow \mathrm{X}a$
end

Algorithm 2: One bit Quantum Bit String Comparator (1BITGQBSC)

The above is a description of the formation of a generalized GQBSC circuit. The mean circuit size grows and shrinks on the basis of input lengths. While Fig. 12 illustrates a circuit for 1-bit GQBSC requiring 4 qubits, Fig. 13 shows the pattern by which it grows for a two-bit size, having a requirement of six qubits. Here, a_0 and a_1 are used for storing the information of the first number, while b_0 and b_1 are used for storing the second number. The 1-bit Comparator is successively applied to Qubits a_0 and b_0, and then to a_1 and b_1 after Z-measurements. Similar representation are provided for 3-bit (Fig. 14) and 5-bit (Fig. 15) inputs.

4 Results

Quantum computing relies on precise measurements for computation outcomes. Traditionally,

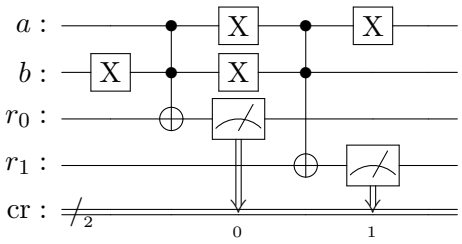

Figure 12: Generalized Quantum Bit String Comparator (GQBSC) Circuit for two inputs with a maximum size of 1 bit

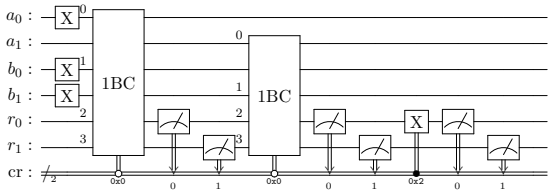

Figure 13: Generalized Quantum Bit String Comparator (GQBSC) Circuit for two-bit size.

measurements were only done at the end to prevent errors. IBM's new hardware-based approach allows dynamic circuits and mid-computation measurements, offering three key benefits; reduced qubit usage, fewer additional operations, and improved accuracy [2].

Given our model, we now present a discussion on its validation, verification, and behavior, all of which are tested with respect to different numbers of input sizes. The Verification of each test case is carried out from single-bit up to 1000-bits; a portion of which is illustrated in Table. 1.

Validation of the n-bit GQBSC model is established on the metrics of quantum cost, quantum delay, and ancillary bit quantity. These are compared against equivalent QBSC circuits of Wang et. al. [22], Al-Rabadi et. al. [23], Thapliyal et. al. [24], Vudadha et. al. [25], David et. al. [26], and Xia et. al. [27]. A comparative analysis of these methods is presented in Table. 4. Here, methods 1-5 are dependent linearly on the input length for the allocation of ancillary qubits, whereas in our case, they remain fixed. This is illustrated in (Fig. 16a). Method 6 has a lower ancillary bit count but has a higher quantum cost and longer quantum delays. The quantum cost of our proposed approach is the same as Method 4 if both $|a\rangle$ and $|b\rangle$ are comparable, and slightly larger otherwise by a factor of $(n-1)/2$ (Also illustrated in Fig. 16b). Here, Method 1 demon-

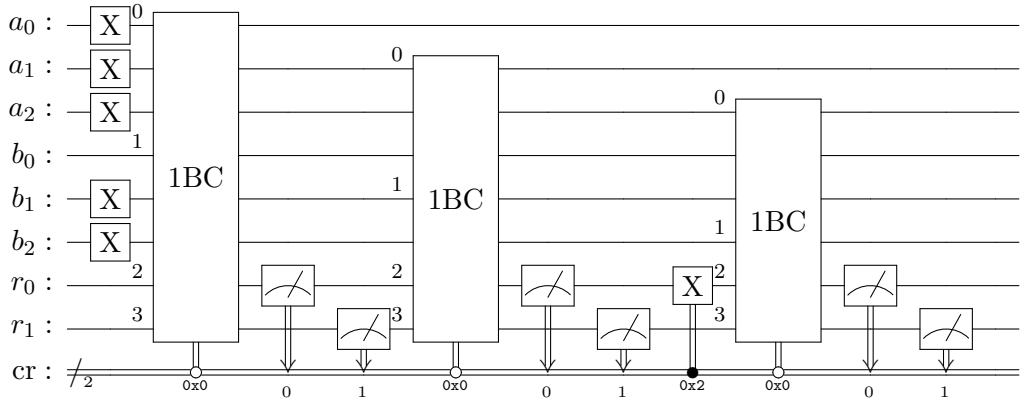

Figure 14: Generalized Quantum Bit String Comparator(GQBSC) Circuit for 3-bit input size.

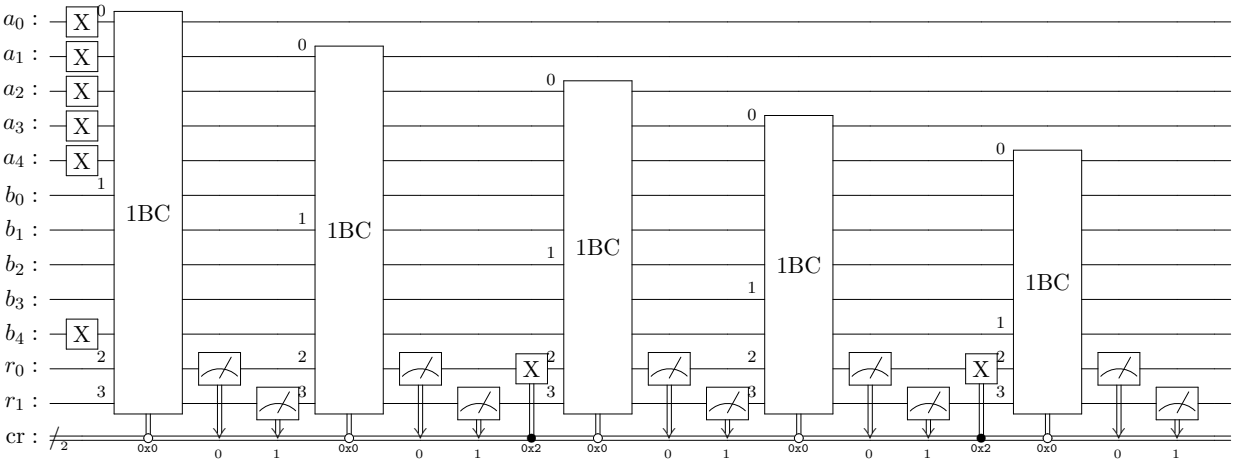

Figure 15: Generalized Quantum Bit String Comparator(GQBSC) Circuit for 5-bit input size.

Sr.	$(\|a\rangle)$	bin $(\|a\rangle)$	$(\|b\rangle)$	bin $(\|b\rangle)$	$(\|a\rangle) \approx (\|b\rangle)$ r_0, r_1	$(\|a\rangle) > (\|b\rangle)$ r_0, r_1	$(\|a\rangle) < (\|b\rangle)$ r_0, r_1	Verification
1	0	0	0	0	00	-	-	verified
2	1	1	0	0	-	01	-	verified
3	0	0	1	1	-	-	10	verified
4	3	11	3	11	00	-	-	verified
5	3	11	1	01	-	01	-	verified
6	1	01	3	11	-	-	10	verified
7	7	111	7	111	00	-	-	verified
8	7	111	3	011	-	01	-	verified
9	3	011	7	111	-	-	10	verified
10	31	11111	31	11111	00	-	-	verified
11	31	11111	30	11110	-	01	-	verified
12	30	11110	31	11111	-	-	10	verified
13	120	1111000	120	1111000	00	-	-	verified
14	127	1111111	63	0111111	-	01	-	verified
15	100	1100100	127	1111111	-	-	11	verified
16	600	1001011000	600	1001011000	00	-	-	verified
17	700	1010111100	420	0110100100	-	01	-	verified
18	630	1001110110	800	1100100000	-	-	11	verified
19	1500	10111011100	1500	10111011100	00	-	-	verified
20	1400	10101111000	200	00011001000	-	01	-	verified
21	560	01000110000	1137	10001110001	-	-	11	verified

Table 1: Verification and Demonstration of various Input Sizes and Expected Outcome of GQBSC on Simulators(*statevector* and *qasm*)

strates exponential cost while the remaining are in linear order. The quantum delay of the proposed approach is illustrated in (Fig. 17c). Here, method 6 exhibits the highest delays due to the complexity of the comparator design. This is followed by methods 2, 5, and 6 in linear order. The delay of method 1 at the start is close to methods 3 and 4 but as the size increases its delay grows linearly. Our proposed method shows the least delay amongst the linear ordered methods. The least delay is noticeable for methods 3-4 which are of logarithmic order. For small input sizes, our proposed method presents a relatively close delay to that of methods 3-4, but for small input sizes only.

The GQBSC circuit was tested on IBM System (*ibm_perth* and *ibm_lagos*). The details of the input tests conducted on these systems and their resulting state probabilities are presented in Table 3. Given that these systems are equipped with a limited number of 7 qubits, the testing was conducted for a maximum of three-bit numbers.

The table is divided into two sections: one for the results obtained on *ibm_perth* and the other for *ibm_lagos*. It is evident that the results of our proposed GQBSC on *ibm_perth* outperform those on *ibm_lagos*. The proposed circuit was executed for 1024 shots for each input, and the probabilities of the four states are presented in the table.

During the testing, it was observed that as we move toward higher qubit numbers, the output quality begins to deteriorate, and the quantum system generates incorrect outputs.

The comparator design is further evaluated through the number of gate operations and circuit complexity and tested against bitstring inputs of various sizes. This is presented in Fig. 17. Here, (a) presents the design with respect to the number of X, CCX, 1BC, and Measurement gates, (b) presents the number of qubits and accumulative width of the quantum and classical registers. Fig. 17c represents the run-time duration when executed on a simulator. Notably, it demonstrates a relatively constant runtime at the outset, but as n grows larger, it exhibits an increase.

The resource analysis reveals that the proposed quantum comparator n qubit for holding the information and 2 ancillary qubit for storing the re-

Sr.	Output State r_0, r_1	Interpretation	Remarks
1	00	$(\|a\rangle) \approx (\|b\rangle)$	`Input-1 == Input-2`
2	01	$(\|a\rangle) > (\|b\rangle)$	`Input-1 > Input-2`
3	10	$(\|a\rangle) < (\|b\rangle)$	`Input-1 < Input-2`
4	11	$(\|a\rangle) < (\|b\rangle)$	`Input-1 < Input-2`

Table 2: Interpretation of GQBSC Output States

Sr.	$(\|a\rangle)$ Input-1	$(\|b\rangle)$ Input-2	$(\|a\rangle) \approx (\|b\rangle)$ $State(0,0)$	$(\|a\rangle) > (\|b\rangle)$ $State(0,1)$	$(\|a\rangle) < (\|b\rangle)$ $State(1,0)$	$(\|a\rangle) < (\|b\rangle)$ $State(1,1)$	$HighestProb.$ $State$	$Expected$ $State$	IBM $Machine$
1	0	0	901	44	74	5	901	$True$	
2	0	1	116	10	861	37	861	$True$	
3	0	2	110	190	8	716	716	$True$	
4	0	3	95	218	5	706	706	$True$	
5	1	0	97	726	20	181	726	$True$	
6	1	1	912	31	63	18	912	$True$	
7	1	2	127	172	12	713	713	$True$	
8	1	3	106	241	13	664	664	$True$	ibm_perth
9	2	0	327	462	7	228	462	$True$	
10	2	1	176	389	3	456	456	$True$	
11	2	2	276	505	6	237	505	$False$	
12	2	3	155	638	2	229	638	$True$	
13	3	0	136	310	4	574	574	$True$	
14	3	1	265	349	3	407	407	$True$	
15	3	2	175	606	4	239	606	$True$	
16	3	3	152	657	4	211	657	$False$	
17	0	0	759	87	166	12	759	$True$	
18	0	1	160	31	782	51	782	$True$	
19	0	2	217	154	6	647	647	$True$	
20	0	3	273	142	9	600	600	$True$	
21	1	0	197	712	40	75	712	$True$	
22	1	1	795	83	145	37	795	$True$	
23	1	2	263	202	8	551	551	$True$	
24	1	3	228	209	11	567	567	$True$	ibm_lagos
25	2	0	261	446	4	313	446	$True$	
26	2	1	179	347	31	467	467	$True$	
27	2	2	335	357	7	325	357	$False$	
28	2	3	261	338	9	416	416	$True$	
29	3	0	236	443	10	338	443	$True$	
30	3	1	218	308	35	468	468	$True$	
31	3	2	262	304	14	444	444	$True$	
32	3	3	203	301	31	489	489	$False$	

Table 3: Result of various Input Sizes and Expected Outcome of GQBSC on real Quantum Computer

Method	Reference	Anc. Bits	Quantum Cost	Quantum Delay
1.	Wang et. al. [22]	$2n$	$O(n^2)\triangle$	$O(n^2)$
2.	Al-Rabadi et. al. [23]	$6n+1$	$(39n+9)\triangle$	$24n+9$
3.	Thapliyal et. al. [24]	$4n-3$	$(18n+9)\triangle$	$18\log(2*n)+7$
4.	Vudadha et. al. [25]	$4n-2$	$(14n)\triangle$	$5\log(2*n)+12$
5.	David et. al. [26]	$3n-1$	$(99(n-1)+12)\triangle$	$20n-1$
6.	Xia et. al. [27]	1	$(28n)\triangle$	$31n+2$
7.	Proposed Comparator	2	$14n\triangle; \quad if a == b$ $(14n+(n-1)/2\triangle; \quad if a \neq b$	$4n; \quad if a == b$ $4n+(n-1)/2; \quad if a \neq b$

Table 4: The ancillary qubit, quantum costs, and quantum delay of proposed method against available methods in literature

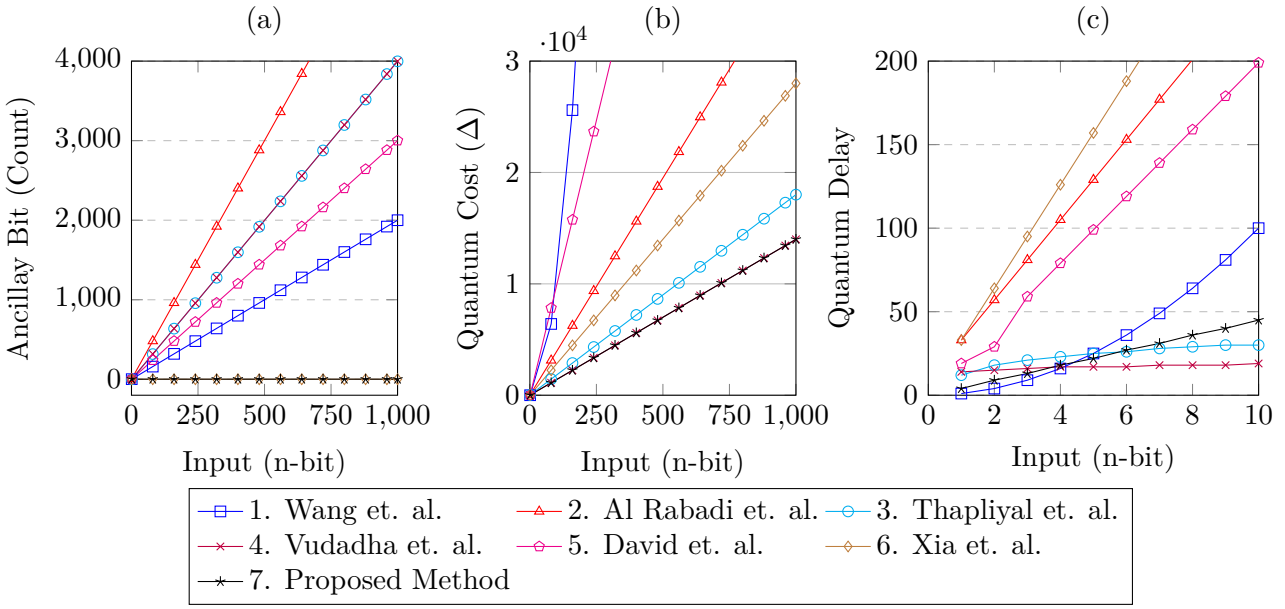

Figure 16: Graphical plot of the number of (a) ancillary bits, (b) quantum cost, and (c) quantum delay of the proposed method against comparable methods of literature.

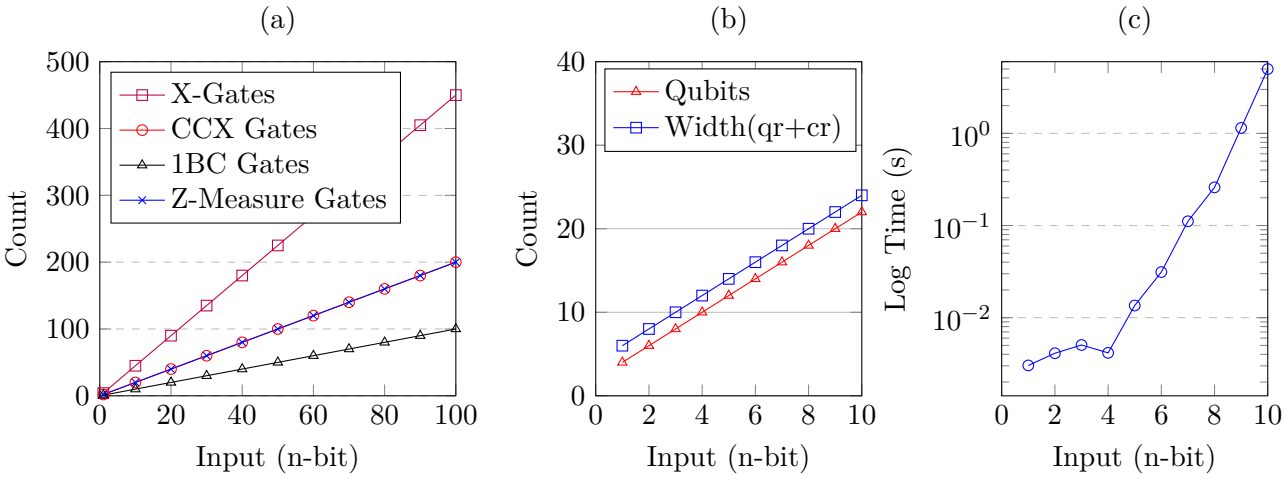

Figure 17: Graphical plot of (a) number of X, CCX, 1BC, and measurement gates, (b) number of Qubits and the width of quantum and classical registers, and (c) run time in seconds, for the proposed method

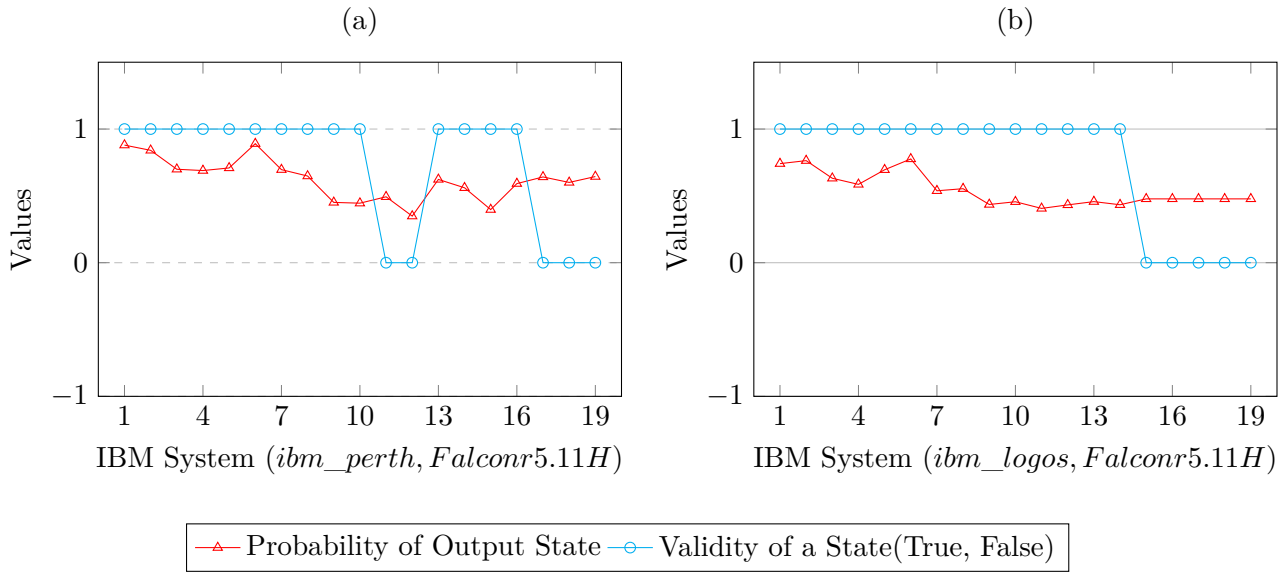

Figure 18: Graphical plot of the number of (a) and (b) show the different result of proposed method on IBM Systems .

sult of comparison, where n represents the size of the binary numbers being compared. This linear scaling suggests that the comparator can handle larger numbers with a manageable increase in resource requirements. The potential advantages of quantum comparators over classical counterparts include parallel processing, reduced computational complexity, and possible speedup in specific applications.

We now present a behavior analysis of our proposed n-bit quantum bit string operator by delineating all the gate operations that it encompasses in the algorithm. The representation allows us to establish a composition-based mathematical representation of the input bitstrings and the quantum output states. This is expressed as in (Equ. 3):

$$
|r_0\rangle = \begin{cases} |r_0\rangle_i \oplus (|a_1\rangle_i \wedge |b_0\rangle_i)\,, & \text{if n} = 1 \\ [|r_0\rangle_i \oplus (|a_1\rangle_i \wedge |b_0\rangle_i)] \\ \qquad \oplus \left[|a_1\rangle_{i+1} \wedge |b_0\rangle_{i+1}\right], & \text{if n} > 1, |r_0\rangle_i = |0\rangle\,, |r_1\rangle_i = |0\rangle\,, |r_1\rangle_{i+1} \neq |1\rangle \\ [|r_0\rangle_i \oplus (|a_1\rangle_i \wedge |b_0\rangle_i)] \\ \qquad \oplus [|a_1\rangle_{i+1} \wedge |b_0\rangle_{i+1}] \cdot [\mathrm{X}(|r_0\rangle_{i+1})], & \text{if n} > 1\, |r_0\rangle_i = |0\rangle\,, |r_1\rangle_i = |0\rangle\,, |r_1\rangle_{i+1} = |1\rangle \end{cases}
$$

$$
|r_1\rangle = \begin{cases} |r_1\rangle_i \oplus (|a_0\rangle_i \wedge |b_1\rangle_i)\ , & \text{if n} = 1 \\ [|r_1\rangle_i \oplus (|a_0\rangle_i \wedge |b_1\rangle_i] \\ \qquad \oplus \left[|a_1\rangle_{i+1} \wedge |b_0\rangle_{i+1}\right], & \text{if n} > 1, |r_0\rangle_i = |0\rangle\,, |r_1\rangle_i = |0\rangle\,, |r_1\rangle_{i+1} \neq |1\rangle \end{cases} \tag{3}
$$

where n is the bit string length, $|r_0\rangle$ and $|r_1\rangle$ contain the final results of the comparator, $i < n$ is the current iteration, and consequently $i+1$ is the next iteration, and any $|\psi\rangle_i$ represents the intermediate results of a quantum state with respect to the iterator i. The final output $|r_0\rangle$ and $|r_1\rangle$ is conditional and depends on the bit string length, and the intermediate result states approximation

to either $|0\rangle$ or $|1\rangle$. By applying these conditions, we can achieve the same outcome as that obtained from our proposed method discussed in Algo. 1 or illustrated in Figs. 12-15.

In Fig. 18, we have included some statistics of the proposed GQBSC method applied to the IBM System (*ibm_perth* and *ibm_lagos*), both of which have a system size of 7 qubits. These graphs clearly illustrate that as the input size approaches the maximum capacity of the qubit system, the performance of the proposed state begins to deteriorate, and the probability of the output state eventually declines for comparable input sizes.

5 Conclusion

Quantum bit string comparators are essential to many quantum algorithms as they provide a means to compare two quantum states. In this study, we have investigated the resource requirements and scalability of our proposed quantum comparator. The analysis demonstrates that our approach efficiently compares input states of varying sizes with an upper bound for resource scaling of n+2 qubits. Holistically, our proposed approach fares better when compared to other methods based on ancillary qubits, quantum cost, and quantum delay. These findings contribute to the ongoing development of quantum algorithms. The proposed approach is extensively tested, verified, and validated against other methods.

References

[1] Yewei Yuan, Chao Wang, Bei Wang, Zhao-Yun Chen, Meng-Han Dou, Yu-Chun Wu, and Guo-Ping Guo, *An Improved QFT-Based Quantum Comparator and Extended Modular Arithmetic Using One Ancilla Qubit*, *arXiv preprint arXiv:2305.09106*, (2023).

[2] F. Hua, Y. Jin, Y. Chen, S. Vittal, K. Krsulich, L. S. Bishop, J. Lapeyre, A. Javadi-Abhari, and E. Z. Zhang, "CaQR: A Compiler-Assisted Approach for Qubit Reuse through Dynamic Circuit," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, 2023, pp. 59–71.

[3] H. Maity, "Design and implementation of a two-qubit quantum comparator circuit (Q-CC)," *Journal of Computational Electronics*, vol. 21, no. 2, pp. 530-534, 2022.

[4] H. M. H. Babu, K. M. M. Uddin, R. B. Himel and N. Biswas, "Quantum Technology for Comparator Circuit," *2022 23rd International Symposium on Quality Electronic Design (ISQED)*, 2022, pp. 1-1.

[5] X. Huang, Y. Chang, W. Cheng, M. Hou and S.-B. Zhang, "Quantum private comparison of arbitrary single qubit states based on swap test," *Chinese Physics B*, vol. 31, no. 4, pp. 040303, 2022, IOP Publishing.

[6] S. M. Ghadamgahi, R. Sabbaghi-Nadooshan and K. Navi, "Novel single-trit comparator circuits in ternary quantum-dot cellular automata," *Analog Integrated Circuits and Signal Processing*, vol. 111, no. 3, pp. 353-370, 2022, Springer.

[7] Zhuang, Xi-Ning and Chen, Zhao-Yun and Xue, Cheng and Wu, Yu-Chun and Guo, Guo-Ping, "Quantum Encoding and Analysis on Continuous Stochastic Process", 2022.

[8] M. Gao, J. Wang, S. Fang, J. Nan and L. Daming, "A new nano design for implementation of a digital comparator based on quantum-dot cellular automata," *International Journal of Theoretical Physics*, vol. 60, pp. 2358-2367, 2021, Springer.

[9] F. Orts, G. Ortega, A. C. Cucura, E. Filatovas and E. M. Garzón, "Optimal fault-tolerant quantum comparators for image binarization," *The Journal of Supercomputing*, vol. 77, pp. 8433-8444, 2021, Springer.

[10] K. Georgopoulos, C. Emary and P. Zuliani, "Comparison of quantum-walk implementations on noisy intermediate-scale quantum computers," *Physical Review A*, vol. 103, no. 2, pp. 022408, 2021, APS.

[11] Nikitas Stamatopoulos, Daniel J Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, Stefan Woerner. *Option pricing using quantum computers*. Quantum, 4:291, 2020. Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[12] Y. Kang and J. Heo, "Quantum minimum searching algorithm and circuit implementation," *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 214-219.

[13] B. Razavi, "The design of a comparator [the analog mind]," *IEEE Solid-State Circuits Magazine*, vol. 12, no. 4, pp. 8-14, 2020.

[14] V. Menon and A. Chattopadhyay, "Quantum string comparison method," *arXiv preprint arXiv:2005.08950*, 2020.

[15] A. Norouzi Doshanlou, M. Haghparast, M. Hosseinzadeh and M. Reshadi, "Efficient design of quaternary quantum comparator with only a single ancillary input," *IET Circuits, Devices and Systems*, vol. 14, no. 1, pp. 80-87, 2020, Wiley Online Library

[16] Haiying Xia, Haisheng Li, Han Zhang, Yan Liang, and Jing Xin, *Novel multi-bit quantum comparators and their application in image binarization*, *Quantum Information Processing*, vol. 18, pp. 1-17, 2019, Springer.

[17] S. S. Roy, C. Mukherjee, S. Panda, A. K. Mukhopadhyay and B. Maji, "Layered T comparator design using quantum-dot cellular automata," *2017 Devices for Integrated Circuit (DevIC)*, 2017, pp. 90-94.

[18] M. Morrison, M. Lewandowski and N. Ranganathan, "Design of a tree-based comparator and memory unit based on a novel reversible logic structure," *2012 IEEE Computer Society Annual Symposium on VLSI*, 2012, pp. 231-236.

[19] Himanshu Thapliyal and Nagarajan Ranganathan, *Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs*, *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 6, no. 4, pp. 1–31, 2010, ACM New York, NY, USA.

[20] S.T. Cheng and C.Y. Wang, *Quantum switching and quantum merge sorting*, IEEE Transactions on Circuits and Systems I: Regular Papers, 53(2), pp. 316-325, 2006.

[21] I. M. Tsai, S. Y. Kuo, and D. S. L. Wei, *Quantum boolean circuit approach for searching and unordered database*, Proc. 2nd IEEE Conf. Nanotechnol., pp. 325-318, August 2002.

[22] Wang, Dong; Liu, Z-H; Zhu, W-N; Li, S-Z. *Design of quantum comparator based on extended general Toffoli gates with multiple targets.* Publisher: Guojia Kexue Jishu Bu, Xinan Xinxi Zhongxin, 132 Shengli Lu Chongqing, China ... *Computer Science*, 39(9), 302–306, 2012.

[23] Al-Rabadi, Anas N. *Closed-system quantum logic network implementation of the Viterbi algorithm. Facta Universitatis-Series: Electronics and Energetics*, 22(1), 1–33, 2009.

[24] Thapliyal, Himanshu; Ranganathan, Nagarajan. *Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs.* ACM Journal on Emerging Technologies in Computing Systems (JETC), 6(4), 1–31, 2010. Publisher: ACM New York, NY, USA.

[25] Vudadha, Chetan and Phaneendra, P Sai and Sreehari, V and Ahmed, Syed Ershad and Muthukrishnan, N Moorthy and Srinivas, Mandalika B. *Design of prefix-based optimal reversible comparator* In *2012 IEEE Computer Society Annual Symposium on VLSI*, pp. 201–206, 2012. Organization: IEEE.

[26] David Sena Oliveira and Rubens Viana Ramos, *Quantum bit string comparator: circuits and applications*, Quantum Comput. Comput, vol. 7, no. 1, pp. 17-26, 2007.

[27] Haiying Xia, Haisheng Li, Han Zhang, Yan Liang, and Jing Xin, *An efficient design of reversible multi-bit quantum comparator via only a single ancillary bit*, *International Journal of Theoretical Physics*, vol. 57, pp. 3727-3744, 2018, Springer.

[28] *Multidimensional color image storage, retrieval, and compression based on quantum amplitudes and phases.* Hai-Sheng Li, Qingxin Zhu, Ming-Cui Li, Hou Ian, and others. *Information Sciences*, vol. 273, pp. 212–232, 2014. Elsevier.

[29] *The quantum search algorithms for all solutions.* Hai-Sheng Li, Qingxin Zhu, Lan Song, and Qian Wu. *International Journal*

of Theoretical Physics, vol. 52, no. 6, pp. 1893–1907, 2013. Springer.