

An Efficient Nearest Neighbor Design for 2D Quantum Circuits



A. Bhattacharjee, C. Bandyopadhyay, B. Mondal, Robert Wille, Rolf Drechsler and H. Rahaman

Abstract In the last couple of years, synthesis of quantum circuits has received huge impetus among the research communities after the evolution of an efficient and powerful computational technology called “quantum computing”. But physical implementation of these circuits considers the nearest neighbor qubit interaction as the desirable one otherwise a computational error can result. Realization of such an architecture in which qubit interacts only with its adjacent neighbors is termed as the *Nearest Neighbor* (NN) property. To attain such design architecture, SWAP gates plays a significant role of bringing the qubits to adjacent locations. But this in turn introduces design overhead so NN-based realization using limited number of SWAP gates has become significant. In order to explore this area, in this article, we introduced an efficient design technique for NN realization of quantum circuits in 2D architecture. The design algorithm has been partitioned into three phases of qubit selection, qubit placement and SWAP gate implementation. To verify the exactness of the stated design approach, its functionality has been evaluated over a wide set of benchmark function and subsequently witnessed an improvement on its cost metrics. By running our algorithm an overall improvement of about 17%, 3% against existing

A. Bhattacharjee (✉) · C. Bandyopadhyay · B. Mondal · H. Rahaman
Indian Institute of Engineering Science and Technology, Shibpur 711103, India
e-mail: anirbanbhattacharjee330@gmail.com

C. Bandyopadhyay
e-mail: chandanb@it.iiests.ac.in

B. Mondal
e-mail: bappa.arya@gmail.com

H. Rahaman
e-mail: rahaman_h@it.iiests.ac.in

R. Wille
Institute for Integrated Circuits, Johannes Kepler University Linz, 4040 Linz, Austria
e-mail: robert.wille@jku.at

R. Drechsler
Institute of Computer Science, University of Bremen & Cyber-Physical Systems, DFKI GmbH,
28358 Bremen, Germany
e-mail: drechsle@uni-bremen.de

2D works and 35%, 22% against 1D works over SWAP gate count and quantum cost metrics have been recorded, respectively.

1 Introduction

Quantum computing, a new computational technology has shown promise in making the computation much easier by overcoming the limitations of conventional computing paradigm. Introduction of such computing technology not only finds solutions for some intractable problems but also solves them within a reasonable time bound.

Due to such computational facilities, quantum computing has left a remarkable footprint in the research community and thereby an effort has been evolved toward the establishment of quantum devices [1]. Additionally, quantum algorithms are required to be designed such that quantum computing architectures can be fabricated using quantum circuits. In quantum paradigm, quantum circuits are designed using a sequence of elementary quantum gates, representing quantum operators that manipulate quantum data information. To this end, qubits are considered as the basic quantum data units that acts similar to bits used in classical computing. However, these quantum units differ from the conventional data units in the way they exist. Unlike classical bits, qubits can be found to occur in multiple states simultaneously which can be represented as the linear superposition of the basis states.

The states of the quantum units are delicate and can easily be modified by environmental effects that can hamper the integrity of the quantum system. Therefore, an essential requirement toward attaining a practical and reliable quantum operation is by conducting fault-tolerant computation. For this purpose, quantum error correction codes turns out to be useful and thereby becomes acceptable [2]. However, application of these codes depends upon the nearest neighbor interaction of the qubits. Additionally, such a restriction in qubit interaction has also been considered as the limiting design constraint for the synthesis of certain quantum implementation technologies like ion trap [3], quantum dots [4], superconducting qubits [5] and nuclear magnetic resonance [6]. To make it more precise, the need toward realization of a nearest neighbor qubit interaction can be warranted due to the limitations of J-coupling force [7] required to enable multi-qubit operations (2-qubit or more) and can only be achieved effectively for adjacent neighboring qubits.

This design architecture can be obtained by making the quantum gates to act only on qubits located at adjacent positions, which can be realized via. SWAP gates. It can be made possible by exchanging the states of the qubits till the desired qubits become adjacent. Realization of NN architectures with the help of SWAP gates in turn causes an impact on the resultant architecture by enhancing the circuit depth and gate count. Therefore, NN optimization in terms of reduction in the number of SWAP gate requirements has become an essential design challenge such that the resulting overhead in the circuit can be checked. To address this, several articles related to efficient NN realization has been declared where the authors mainly considered SWAP gate reduction by following different design solutions. To realize NN

design architecture, qubits need to be mapped onto the desired topological layout structure and the most commonly used structure is the qubit chain or 1D layout. Such an arrangement of qubits is referred to as the Linear Nearest Neighbor (LNN) architecture. Several contributions toward the development of LNN architecture have been made and detailed review of those works is presented next.

The importance of NN design networks along with the discussion of various design methodologies has been stated in the article [8]. In [9], a couple of efficient transformation schemes related to LNN synthesis is introduced that reduces the additional circuit and time complexity of the resultant structures. To produce cost-effective NN architectures, the authors of [10] suggested a transformation approach where mapping of circuits into lattice structure representations has been presented. The use of graph partitioning algorithm to form improved linear NN design architectures is stated in [11]. To improve the design further, two reordering techniques viz. global and local has been discussed in the work [12], where exact NN design solutions for each of the two reordering schemes are introduced so that the resultant solution becomes optimal. In pursuit of having an efficient NN realization, the authors of [13] presented a design methodology based on local reordering approach in which SWAP gate optimization is achieved by exploiting look-ahead strategies. To obtain efficient NN solutions for larger circuits, a heuristic approach using the look-ahead policy has been stated in the work [14]. Furthermore, an optimal linear NN architecture using global-based scheme has been obtained in [15], where an exact design algorithm based on A* approach is employed.

However, the stated research contributions discussed so far are based on LNN synthesis of quantum circuits where the qubits are arranged to form a chain like structure in which it can hardly have more than two adjacent neighbors to communicate. To maximize this nearest neighbor communication, the qubits need to be mapped to higher dimensional topological structures like 2D, 3D, and even on multidimensional layout. In other words, 2D organization of qubits facilitates to communicate with a maximum of four neighbors while such nearest neighbor communication can extend up to six in case of 3D structures. But such an increase in communication makes it difficult in controlling the qubits especially in 3D representations. Therefore, in this work, we have considered only on synthesis of 2D NN architectures. In the recent past, many research articles related to NN representations in 2D layouts have been reported and a few of them are stated below.

In the work [16], a mixed integer programming approach has been undertaken in which the design problem is formulated by mapping it into a 2D architecture and provides better design structures than 1D representation. An optimal 2D solution with respect to SWAP cost is derived in [17] whereby an exact design strategy has been employed. Despite the algorithm produces an optimal structure but it was found infeasible for large benchmark circuits as a result of extensive computational cost. In [18], the authors reduce the design overhead by arranging the qubits in such a manner that they are placed at suitable grid positions. To obtain a better NN representation, a heuristic design scheme has been undertaken in [19] in which a couple of grid selection approaches is employed followed by an efficient qubit placement strategy where the circuit's qubits are mapped based on their corresponding interaction

count. To optimize the design further, the authors of [13] exploited a heuristic-based look-ahead design methodology that determines the appropriate movement of the qubits resulting in a significant reduction in SWAP cost. To provide an improved and scalable 2D NN representation, a better heuristic model based on generalization for a combined local and global reordering scheme has been investigated in the work [20].

Here, in this article we have employed a heuristic qubit mapping strategy to transform a quantum circuit to its corresponding 2D NN representation using less number of additional SWAP gates. Our design methodology has shown better performance over some of the existing research articles.

The remaining content of the article is formulated as follows. In Sect. 2, we discuss about the elementary quantum gates along with the nearest neighbor property. A detailed discussion of our design methodology has been presented in Sect. 3. An experimental evaluation of our approach against the related works has been summarized in Sect. 4. Finally, the article ends with concluding remarks in Sect. 5.

2 Background

In quantum technology, qubits are considered as the elementary quantum data units whose states undergo modification through execution of a sequence of quantum gates. Qubits share a similar characteristic of bits is that it can occur in one of the basis states like $|0\rangle$ and $|1\rangle$, which can be considered equivalent to 1 and 0 in conventional computing. In addition to these basis states, qubits can even occur in superimposed states of $|0\rangle$ and $|1\rangle$ that can be interpreted in the form of a state vector expression as

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$


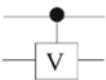



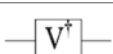
where the notations α and β in the above expression denotes the complex numbers indicating the probability amplitudes of the corresponding basis states that satisfy the condition $\alpha^2 + \beta^2 = 1$. While measurement causes this state vector to degenerate into one of the basis states of $|0\rangle$ and $|1\rangle$.

The operations performed by the quantum gates on the qubits can be defined in the form of unitary matrices. In this context, the quantum functionality of an n -qubit quantum system can be realized via multiplication of distinct $2n \times 2n$ unitary matrices.

Definition 1 Quantum gates represent the elementary quantum operators that manipulate the qubits and when a collection of such gates are arranged over any group of circuit lines then the formed circuit is termed as quantum circuit.

The gates that have been most popularly used in the implementation of a quantum circuit are CNOT, NOT, and $V/V+$ and their corresponding symbolic representation have been represented in Table 1. These quantum gates form the constituent elements of the NCV gate library [21, 22] that help to map specific quantum into

Table 1 Symbolic representations of some quantum gates

Gates	Representation	Gates	Representation
NOT		Controlled-V	
CNOT			
V		Controlled-V†	
V†			

its corresponding gate level representations. In this chapter, our work is formulated using only the quantum gates from NCV library.

Despite the fact that quantum gates can be used to transform any function into a relevant circuit representation but there exist some physical limitations for the implementation of these circuits. To this end, realization of such circuits requires the quantum gates to interact only with its qubits physically placed at adjacent positions. In order to satisfy this design constraint, the process of implementing SWAP gates before any gate with non-neighboring qubits becomes essential that alters the positions of the qubits till the desired qubits become adjacent. Such a requirement for the physical design of quantum circuits is regarded as the Nearest Neighbor condition and it can be further described as follows:

Definition 2 Nearest Neighbor Cost of any 2-qubitgate $g(c, t)$ can be interpreted as the distance separating the positions of its control (c) and target (t) qubits and this difference can be computed mathematically as

$$NNC(g) = |c - t| - 1 \tag{2}$$

The above expression determines the nearest neighbor cost ($NNC(g)$) of an individual gate g and the combination of such costs of the respective gates produces the overall cost ($NNC(QC)$) of the corresponding quantum circuit QC . It can be represented mathematically as follows:

$$NNC(QC) = \sum NNC(g) \tag{3}$$

This interpretation indicates that a given circuit holds the nearest neighbor condition provided either all the 2-qubit quantum gates act on adjacent qubits or it is having only 1-qubit gates. In other words, if the nearest neighbor expression represented above evaluates to zero then the resultant circuits are said to be NN-compliant.

Fig. 1 **a** Toffoli gate structure, **b** NCV realization of Fig. 1a

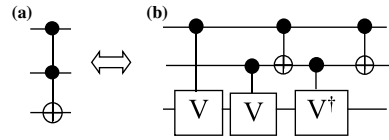
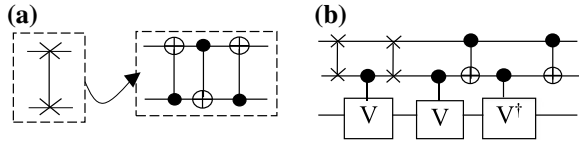


Fig. 2 **a** SWAP gate, **b** NN structure of Fig. 1b



Suppose, consider a Toffoli gate represented in Fig. 1a, whose corresponding NCV gate level realization obtained after decomposition is depicted in Fig. 1b. By inspecting Fig. 1b, it can be observed that the NCV realization of Toffoli gate does not hold the nearest neighbor condition as its overall NNC holds a positive ($Q_C = 1$).

To make the circuit as NN-compliant, SWAP gates (diagrammatically represented in Fig. 2a) needs to be inserted before the first gate with nonadjacent interacting qubits which changes the qubit positions till they are placed adjacent. The resulting NN circuit post SWAP injection is depicted in Fig. 2b.

To get a clear understanding of the purpose of SWAP gate, another example has been considered in which transformation of a non-NN-complaint into its corresponding NN representation using SWAP gates is illustrated.

Example 1 Let's consider the circuit as depicted in Fig. 3a that fails to meet the nearest neighbor criteria as all the 2-qubit gates does not have their interacting qubits placed adjacent. SWAP gates are needed to transform the given circuit to its equivalent NN design. The resultant NN circuit is obtained after inserting ten SWAP gates as depicted in Fig. 3b.

The transformation process we have discussed so far is related to NN architectural design of 1D quantum circuit. It is also possible to obtain a much better NN representation for the corresponding circuit by projecting it in a 2D architectural format in which the qubits are mapped from a linear chain like structure to a two-dimensional

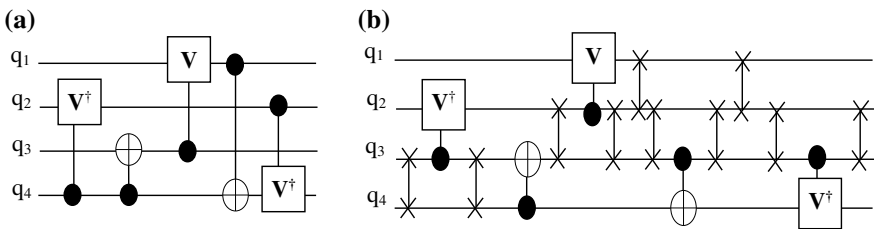


Fig. 3 **a** Quantum circuit with $NNC = 5$, **b** NN-compliant representation of Fig. 3a

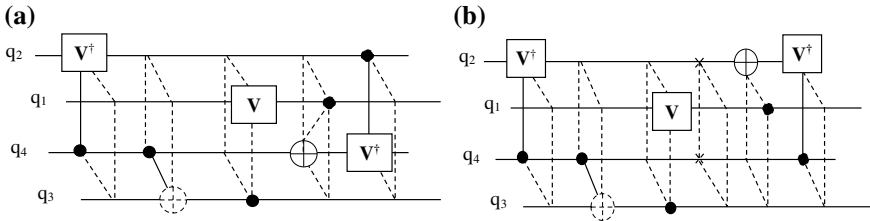


Fig. 4 a Orientation of circuit in Fig. 3a into 2D structure, b NN representation of Fig. 4a

grid structure. Such a mapping process produces an improved NN architecture over its linear counterpart by reducing the SWAP gate requirement. Now, we will discuss about the NN representation of quantum circuits in 2D architecture. The transformation process of quantum circuit into a two-dimensional topological layout can be carried out by mapping the qubits from linear format into a grid-like structure whereby the qubits are permitted to interact with four adjacent neighbors while only two nearest neighbor interaction can be allowed in 1D layout. Such an increment in nearest neighbor communication of qubits minimizes the number of nonadjacent gates and thereby leads to an efficient representation of NN structure with fewer SWAPS overhead in the circuit. Moreover, optimization of the resultant design structures depends on the selection of an appropriate 2D configuration grid used in the qubit mapping process.

Consider the following mapping of a five-qubit circuit into a 2D structure. There exist several possible solutions in which such a mapping process can be conducted depending upon the availability of grid configurations. Likewise, a five-qubit circuit can be represented in 2×3 , 3×2 , or 3×3 .

Example 2: Considering the circuit shown in Fig. 3a, transformed into 2D structure in which a 2×2 configuration is chosen for arranging the qubits randomly as depicted in Fig. 4a. After inserting the necessary SWAP gates the resultant NN circuit realized in 2D topological layout is shown in Fig. 4b.

3 Proposed Approach

In this chapter, an improved heuristic qubit mapping scheme for the synthesis of NN circuits in 2D configuration has been described. This design workflow realizes an efficient NN architecture in which the circuit overhead is reduced by controlling SWAP gate implementation. For this purpose, our synthesis mechanisms are developed based on some heuristic policy that can be used for making some design-oriented decision and thereby formulated a unique qubit placement strategy for better mapping of qubits.

In this regard, our design workflow has been segmented into three phases namely qubit selection, followed by qubit placement and then SWAP gate insertion. For better realization, all the aforementioned stages have been explained with an illustrative example and for this any circuit specification such as the one shown in Fig. 5a is considered as the input on which all the desired operations pertaining to the synthesis approach are conducted.

Phase1: Qubit Selection Policy

The purpose of this phase is to arrange the qubits in a suitable manner on the 2D grid structure. To fulfill this objective, we have used some qubit preference metrics in which two metric tables viz. *time interaction* and *time costing* are computed by reading the gate level specifications of the given circuit.

Total interaction time of each individual qubit is estimated and recorded the results in a *time interaction* table. The values stored in this table constitute the interacting timestamps of the operating qubits and their corresponding overall interaction time is determined by aggregating the timestamps for each individual qubits in the given circuit. For the circuit shown in Fig. 5a, its *time interaction* table is estimated and represented in Table 2.

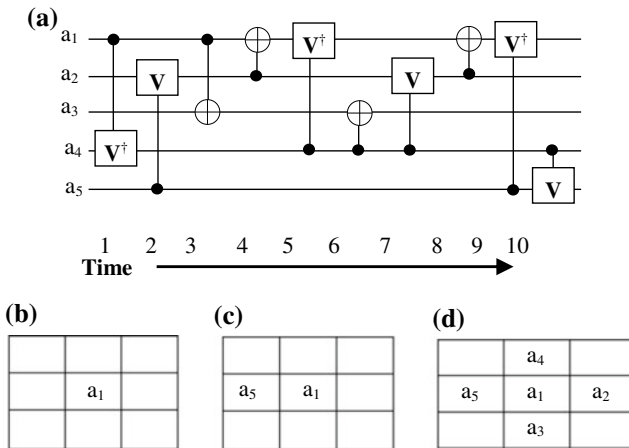


Fig. 5 a Input circuit, b a₁ inserted in the center of 3 × 3 grid, c a₅ placed on the left of a₁, d Qubits a₄, a₂, a₃ placed around a₁

Table 2 Qubit time interaction table

Qubits	Time instants	Total interaction
a ₁	1, 3, 4, 5, 8, 9	30
a ₂	2, 4, 7, 8	21
a ₃	3, 6	9
a ₄	1, 5, 6, 7, 10	29
a ₅	2, 9, 10	21

Table 3 Qubit time costing table

Qubits	Time instants	Total interaction	Time instants	Total costing
a ₁	1, 3, 4, 5, 8, 9	30	1, 3, 5, 9	18
a ₂	2, 4, 7, 8	21	2, 7	9
a ₃	3, 6	9	3	3
a ₄	1, 5, 6, 7, 10	29	1, 5, 7	13
a ₅	2, 9, 10	21	2, 9	11

Table 4 Qubit preference table

Qubits	Total interaction	Total costing	Preference index
a ₁	30	18	18/30 = 0.6
a ₂	21	9	9/21 = 0.42
a ₃	9	3	3/9 = 0.33
a ₄	29	13	13/29 = 0.44
a ₅	21	11	11/21 = 0.52

After resolving the qubit interaction metric discussed above then we work on the time-related cost of all the interacting qubits in the circuit. Computation of this cost parameter involves evaluation of the total costing time of the qubits and for this calculation purpose, the previous qubit interaction (*time interaction*) table determined is needed. Finally, the computation ends by recording the qubit costing results in a table called *time costing* (see Table 3). This parameter is determined by identifying the time steps in which a qubit of any 2-qubit gate is not interacting with an adjacent neighboring qubit using the respective qubit interaction time values and in this manner, such timestamp evaluation is carried out for all the individual qubits of the circuit provided.

After deriving these two tables, we merge the information contained in them and stored the combined result in a new table called qubit preference table is tabulated in Table 4. This preference table contains the qubit preference index evaluated using the ratio between the costing and interaction time values for all the qubit acting as inputs in a given circuit. Next, this preference table is sorted in decreasing order using the qubit index values computed earlier in which the qubit with largest indexing value is stored at the beginning of the table as displayed in Table 5. Now the index entries in the sorted preference table represent the qubit priority values used in the decision making purpose related to qubit placement as discussed in the next phase.

Phase2: Qubit Placement Policy

At the end of the previous workflow process, a preference table providing the sequence to be followed for qubits mapping on a 2D grid structure is considered as the input of this phase on which our mapping algorithm is executed as discussed next.

Table 5 Sorted qubit preference table

Qubits	Total interaction	Total costing	Preference index
a ₁	30	18	0.6
a ₅	21	11	0.523
a ₄	29	13	0.448
a ₂	21	9	0.428
a ₃	9	3	0.33

The qubit mapping process works by picking the appropriate qubits from the preference table (PT) (see Table 5) based on their preference index values and thereby arranges the qubits on the chosen grid position. To be precise, the process starts by selecting the qubit with highest priority index from PT table and positioned the corresponding qubit at the center of the grid structure. After allocating the position of this qubit then a searching process is applied on the resultant grid to look for an empty cell having maximum number of adjacent vacant locations. After detection of such a desired position, the algorithm selects the next preferred qubit from preference table and places it in the identified location. If our search results does not generate a unique vacant cell then the qubits from PT table are placed in these locations by selecting them in the order of left, top, right and bottom with respect to the position of the last placed qubit depending on the space availability and carried out till a definite location is found. Following this algorithmic policy, ordered qubits from preference table are selected and settled on the 2D grid structure.

Now, this same mapping function is employed on the preference table generated in the previous phase to arrange the qubits on a proper 2D network. As a result, the qubit with high priority, a₁ is taken and placed at the center of the chosen grid (3 × 3) as shown in Fig. 5b. After placing qubit a₁, we initiate a search to look for cells surrounded with large number of adjacent empty locations on the corresponding grid. In this case, we have determined eight such possible cells surrounded by a maximum of two vacant cells and to solve this the remaining qubits from the preference table are selected orderly and placed by following the convention like left, top, right and bottom of the last qubit a₁ till a unique empty cell is identified. Hence, the next ordered qubit a₅ in PT is fetched and then placed at a location occurring to the left of a₁ as represented in Fig. 5c. In this manner, the remaining qubits viz. a₄, a₂, a₃ are mapped to the positions locating at the top, right, and bottom of a₁, respectively. The resultant grid structure obtained after placing all the qubits appear as shown in Fig. 5d.

Phase3: SWAP gate insertion

In the previous design phase, our mapping function has organized all the qubits on the given grid and thereby the circuit has been transformed into a 2D representation. Now, we consider the resultant 2D grid obtained in phase 2 and examine the gates with their qubits arranged on a grid and in this process if we identify a one with

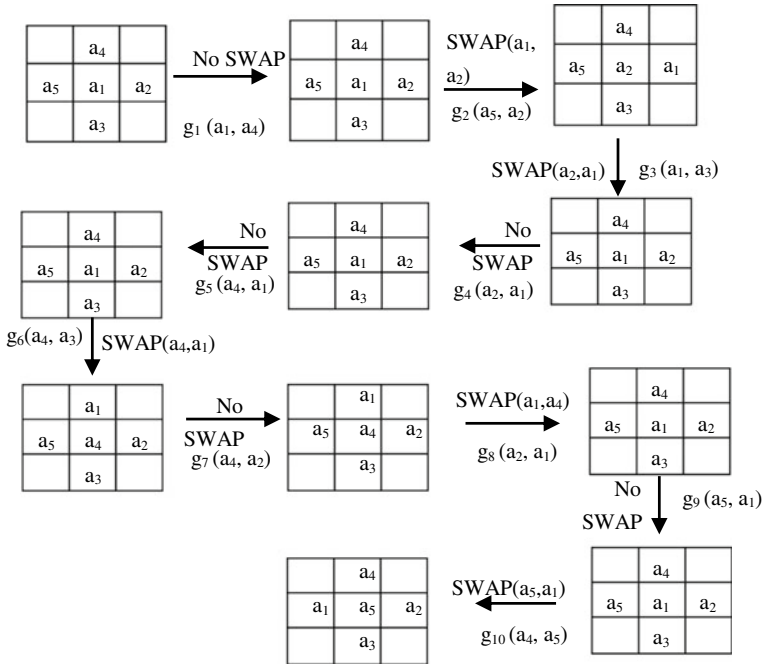


Fig. 6 Steps of SWAP gate insertion

nonadjacent qubits then a SWAP gate is applied to bring those qubits at adjacent positions.

Considering the circuit shown in Fig. 5a, execution of phase 2 generates the structure represented in Fig. 5d and now in this phase we work on this grid and apply SWAP gates at required positions so that an equivalent NN circuit is formed as depicted in Fig. 6. It can be noticed that overall of five SWAP gates are needed to transform the circuit into its corresponding NN form.

To have a better interpretation of the entire design mapping process, we have provided another illustrative example.

Example 3 Here, we have considered a benchmark function, 4gt11_84 as shown in Fig. 7, to describe the transformation mechanism used in our qubit mapping process is represented from Tables 6, 7, 8, 9, 10 (Figs. 8, 9).

4 Experimental Results

The qubit mapping algorithm has been developed using C and executed the function on a machine having Intel i5 processor with 4 GB RAM and 3.30 GHz clock. The performance analysis of our mapping scheme has been made by conducting experi-

Fig. 7 Input benchmark circuit (4gt11_84)

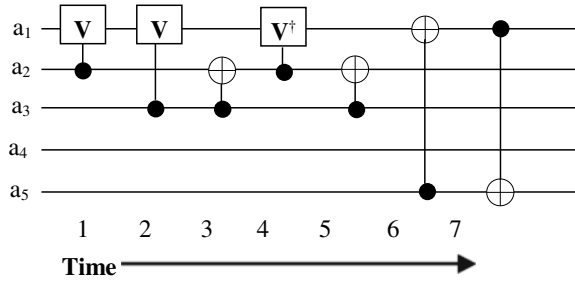


Table 6 Qubit time interaction table

Qubits	Time instants	Total interaction
a ₁	1,2,4,6,7	20
a ₂	1,3,4,5	13
a ₃	2,3,5	10
a ₄	–	–
a ₅	6,7	13

Table 7 Qubit time costing table

Qubits	Time instants	Total interaction	Time instants	Total costing
a ₁	1,2,4,6,7	20	2,6,7	15
a ₂	1,3,4,5	13	0	0
a ₃	2,3,5	10	0	0
a ₄	–	–	–	–
a ₅	6,7	13	6,7	13

Table 8 Qubit preference table

Qubits	Total interaction	Total costing	Preference index
a ₁	20	15	15/20 = 0.75
a ₂	13	0	0/13 = 0
a ₃	10	0	0/10 = 0
a ₄	–	–	–
a ₅	13	13	13/13 = 1

Table 9 Sorted preference

Qubits	Total interaction	Total costing	Preference index
a ₅	13	13	1
a ₁	20	15	0.75
a ₂	13	0	0
a ₃	10	0	0
a ₄	–	–	–

Table 10 Comparison results over medium size benchmark functions

Benchmark's name	Gate count	No. of qubits present	1D results from the work of [16]		2D Results from the work of [16]			2D Results from the work of [19]			Proposed work results		
			Quantum cost	SWAP count	Grid size	Quantum cost	SWAP count	Grid size	Quantum cost	SWAP count	Grid size	SWAP count	Quantum cost
3_17_13	14	3	17	4	2 × 2	20	6	2 × 2	17	3	2 × 2	5	19
4_49_17	32	4	45	12	2 × 2	45	13	-	-	-	2 × 2	10	42
aj-e11_165	60	4	96	36	2 × 3	84	24	3 × 2	82	22	2 × 3	18	78
decod24-v3_46	9	4	12	3	3 × 2	12	3	-	-	-	2 × 2	2	11
hwb4_52	23	4	33	10	2 × 2	32	9	2 × 2	32	9	2 × 2	7	30
rd32-v0_67	8	4	10	2	2 × 3	10	2	-	-	-	2 × 3	2	10
4gt11_84	7	5	8	1	2 × 3	9	2	2 × 3	9	2	2 × 3	2	9
4gt10-v1_81	36	5	56	20	3 × 2	52	16	3 × 2	51	15	2 × 3	14	50
4gt12-v1_89	53	5	88	35	3 × 2	72	19	2 × 4	71	18	3 × 2	20	73
4mod5-v1_23	24	5	33	9	2 × 3	35	11	3 × 2	31	7	2 × 3	8	32
4gt5_75	22	5	34	12	3 × 3	30	8	2 × 5	32	10	3 × 3	9	31
4gt4-v0_80	44	5	78	34	2 × 3	61	17	4 × 4	59	15	2 × 3	16	60
4mod7-v0_95	40	5	61	21	3 × 3	53	13	2 × 5	54	14	3 × 3	10	50
alu-v4_36	32	5	50	18	2 × 3	42	10	2 × 5	43	11	2 × 3	11	43
hwb5_55	109	5	172	63	3 × 2	154	45	2 × 7	158	49	3 × 2	38	147
QFT5	10	5	16	6	3 × 2	15	5	4 × 2	15	5	3 × 2	5	15

(continued)

Table 10 (continued)

Benchmark's name	Gate count	No. of qubits present	1D results from the work of [16]		2D Results from the work of [16]			2D Results from the work of [19]			Proposed work results		
			Quantum cost	SWAP count	Grid size	Quantum cost	SWAP count	Grid size	Quantum cost	SWAP count	Grid size	SWAP count	Quantum cost
hwb6_58	146	6	264	118	2 × 3	225	79	2 × 3	222	76	2 × 3	63	209
mod5adder_128	87	6	138	51	3 × 2	128	41	2 × 3	123	36	3 × 2	28	115
mod8-10_177	109	6	181	72	3 × 3	154	45	4 × 3	152	43	3 × 3	39	148
QFT6	15	6	27	12	2 × 3	21	6	3 × 2	22	7	2 × 3	5	20
rd53_135	78	7	144	66	5 × 2	117	39	2 × 7	118	40	3 × 3	29	107
ham7_104	87	7	155	68	3 × 3	135	48	2 × 7	132	45	3 × 3	38	125
QFT7	21	7	47	26	2 × 4	39	18	6 × 2	35	14	2 × 4	14	35
QFT8	28	8	61	33	4 × 2	46	18	4 × 2	51	23	4 × 2	18	46
QFT9	36	9	90	54	3 × 3	70	34	5 × 2	72	36	3 × 3	24	60
rd73_140	76	10	132	56	4 × 3	113	37	3 × 6	119	43	4 × 3	28	104
sys6-v0_144	62	10	121	59	4 × 4	93	31	-	-	-	4 × 3	30	92
QFT10	45	10	115	70	5 × 3	98	53	4 × 3	96	51	4 × 3	33	78
rd84_142	112	15	260	148	5 × 3	166	54	4 × 5	174	62	5 × 3	48	160
cnt3-5_180	125	16	252	127	3 × 6	194	69	4 × 4	209	84	3 × 6	54	179
Total			2796	971		2325	775		2179	740		628	2178

Fig. 8 Qubits arranged on grid (2×3)

a_2	a_1	a_3
	a_5	a_4

mental evaluations over a set of various benchmark functions taken from [23]. The experimental data set has been summarized in two result tables Tables 10 and 11 respectively. The result sets for small and medium size benchmark specifications are recorded in Table 10 while the other table contains the data of large size functions. In each of these tables, two cost metric values namely quantum cost and SWAP cost are evaluated for all the benchmark functions and compared the estimated results against some existing 1D and 2D design approaches.

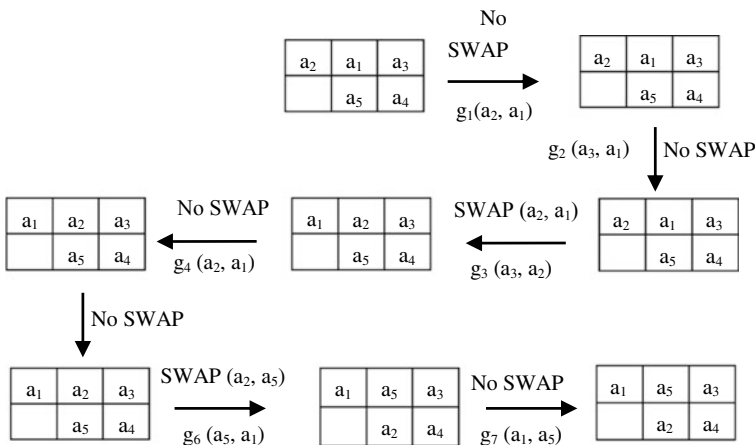


Fig. 9 Stages of SWAP insertion policy

Table 11 Comparison over higher size benchmarks

Benchmark's name	Initial QC incurred	Total lines present	Gate count	Grid size	QC in NN design	No. of SWAPs
rev_17	136	17	136	6×3	443	214
hm_20	73	20	73	5×4	142	69
ac_21_1	130	21	130	6×4	246	116
rev_18	153	18	153	5×4	374	221
ac_21_2	67	21	67	6×4	120	53
rev_19	171	19	171	4×5	427	256
hm_21	79	21	79	6×4	181	102
ac_21_3	42	22	42	6×4	93	51
hm_22	85	22	85	6×4	179	94

Table 10 contains a comparison analysis of our mapping algorithm with the results of the reported work [19] and against the best results of [16]. From this comparison, our proposed mapping approach has shown better performance over the reported works.

From the analysis of the result tables, an overall improvement of about 28.71% and 10.18% with respect to the previous 1D and 2D works has been estimated. In addition to this, a best case improvement of about 37.73% and 35.71% is noticed over the reported 2D articles [16] and [19] whereas an improvement of about 67.56% is attained in case of 1D [16] representation. Investigation of our result tables suggests that our design algorithm provides an improved resultant structure for majority of the benchmark functions. But for few circuits, our mapping approach has not provided a better NN realization than the reported ones.

5 Conclusion

In the present work, an improved heuristic qubit mapping scheme for NN realization of 2D quantum circuits is discussed. The design algorithm has been divided into three segments starting with qubit selection process followed by placement strategy and then SWAP insertion policy. To justify the functionality of our mapping scheme, it has been evaluated over a various set of benchmark circuits and has shown improved results. The computed values are compared with some of the existing 1D and 2D research articles. Based on our evaluation, it can be inferred that the proposed design mapping scheme attains a significant improvement around 35.32%, 22.10% over SWAP and quantum cost metric in 1D whereas an improvement of 17.09%, 3.28% over the same metrics is registered in case of 2D. In spite of having an improvement over the existing articles but the computed results may not be optimal because of implementing heuristic design policy in qubit mapping process and thereby in the future we work on optimizing the design structure further by investigating more efficient design mapping workflow.

References

1. Simonite T (2015) IBM shows off a quantum computing chip. <http://www.technologyreview.com/news/537041/ibm-shows-off-a-quantum-computing-chip/>
2. Chow JM, Gambetta JM, Magesan E, Abraham DW, Cross AW, Johnson BR, Masluk NA, Ryan CA, Smolin JA, Srinivasan SJ, Steffen M (2014) Implementing a strand of a scalable fault-tolerant quantum computing fabric. *Nat Commun* 5:4015
3. Kielpinski D, Monroe C, Wineland DJ (2002) Architecture for a large-scale ion-trap quantum computer. *Nature* 417(6890):709
4. Taylor JM, Petta JR, Johnson AC, Yacoby A, Marcus CM, Lukin MD (2007) Relaxation, dephasing, and quantum control of electron spins in double quantum dots. *Phys Rev B* 76(3):035315

5. Blais A, Gambetta J, Wallraff A, Schuster DI, Girvin SM, Devoret MH, Schoelkopf RJ (2007) Quantum-information processing with circuit quantum electrodynamics. *Phys Rev A* 75(3):032329
6. Criger B, Passante G, Park D, Laflamme R (2012) Recent advances in nuclear magnetic resonance quantum information processing. *Philos Trans Roy Soc A: Math Phys. Eng Sci* 370(1976):4620–4635
7. Grigorenko IA, Khveshchenko DV (2005) Single-step implementation of universal quantum gates. *Phys Rev Lett* 95(11):110501
8. Saeedi M, Wille R, Drechsler R (2011) Synthesis of quantum circuits for linear nearest neighbor architectures. *Quantum Inf Process* 10(3):355–377
9. Hirata Y, Nakanishi M, Yamashita S, Nakashima Y (2011) An efficient conversion of quantum circuits to a linear nearest neighbor architecture. *Quantum Inf. Comput.* 11(1&2):142–166
10. Perkowski M, Lukac M, Shah D, Kameyama M (2011) Synthesis of quantum circuits in linear nearest neighbor model using positive davio lattices. *Facta universitatis-series, Electronics and Energetics*
11. Chakrabarti A (2011) Sur-Kolay, S., Chaudhury, A.: Linear nearest neighbor synthesis of reversible circuits by graph partitioning. [arXiv:1112.0564](https://arxiv.org/abs/1112.0564)
12. Wille R, Lye A, Drechsler R (2014) Exact reordering of circuit lines for nearest neighbor quantum architectures. *IEEE Trans Comput-Aided Des Integr Circuits Syst* 33(12):1818–1831
13. Wille R, Keszocze O, Walter M, Rohrs P, Chattopadhyay A, Drechsler R (2016) Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits. In: Design automation conference (ASP-DAC), pp 292–297
14. Kole A, Datta K, Sengupta I (2016) A heuristic for linear nearest neighbor realization of quantum circuits by SWAP gate insertion using n-gate lookahead. *IEEE J Emerg Sel Topics Circuits Syst* 6(1):62–72
15. Zulehner A, Gasser S, Wille R (2017) Exact global reordering for nearest neighbor quantum circuits using A*. In: International conference on reversible computation. Springer, Cham, pp 185–201
16. Shafaei A, Saeedi M, Pedram M (2014) Qubit placement to minimize communication overhead in 2D quantum architectures. In: Design automation conference (ASP-DAC), pp 495–500
17. Lye A, Wille R, Drechsler R (2015) Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits. In: 2015 20th Asia and South Pacific design automation conference (ASP-DAC), pp 178–183
18. Alfailakawi MG, Ahmad I, Hamdan S (2016) Harmony-search algorithm for 2D nearest neighbor quantum circuits realization. *Expert Syst Appl* 61:16–27
19. Shrivastwa RR, Datta K, Sengupta I (2015) Fast qubit placement in 2D architecture using nearest neighbor realization. In: 2015 IEEE international symposium on nanoelectronic and information systems (iNIS), pp 95–100
20. Kole A, Datta K, Sengupta I (2018) A new heuristic for n-dimensional nearest neighbor realization of a quantum circuit. *IEEE Trans Comput-Aided Des Integr Circuits Syst* 37(1):182–192
21. Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, Shor P, Sleator T, Smolin JA, Weinfurter H (1995) Elementary gates for quantum computation. *Phys Rev A* 52(5):3457
22. Miller DM, Wille R, Sasanian Z (2011) Elementary quantum gate realizations for multiple-control Toffoli gates. In: 2011 41st IEEE international symposium on multiple-valued logic (ISMVL), pp 288–293
23. Wille R, Große D, Teuber L, Dueck GW, Drechsler R (2008) RevLib: an online resource for reversible functions and reversible circuits. In: 2008 38th International Symposium on Multiple Valued Logic, ISMVL, pp 220–225