A Novel Approach for Nearest Neighbor Realization of 2D Quantum Circuits

Anirban Bhattacharjee[1], Chandan Bandyopadhyay[1], Robert Wille[2], Rolf Drechsler[3], Hafizur Rahaman[1]
[1]Indian Institute of Engineering Science and Technology Shibpur, India-711103
[2]Institute for Integrated Circuits, Johannes Kepler University Linz, A-4040 Linz, Austria
[3]Institute of Computer Science, University of Bremen & Cyber-Physical Systems, DFKI GmbH, 28358 Bremen, Germany
Email: anirbanbhattacharjee330@gmail.com, chandanb@it.iiests.ac.in, robert.wille@jku.at, drechsle@uni-bremen.de, rahaman_h@it.iiests.ac.in

***Abstract*— Since decades, quantum computing has received tremendous attention among the researchers due to its dominance over classical computing. But simultaneously it has faced some design challenges and implementation constraints in this long run. One such constraint to build quantum circuits is to satisfy the so-called *Nearest Neighbor* (NN) property in the implemented circuits. Using SWAP gates, this constraint can be satisfied. But this leads to another design issue, namely how to determine such NN designs with a minimum use of SWAP gates. In way to further explore this area, in this work, we propose a heuristic approach for efficient NN complaint representation of quantum circuits in 2D space. The developed technique is segmented in three stages – qubit selection, qubit placement and SWAP gate insertion. The stated approach has been tested over a wide spectrum of benchmarks and reductions in cost parameters are observed. Improvement of more than 17%, 3% over 2D designs and 35%, 22% over 1D designs on SWAP count and quantum cost can be reported, respectively.**

Keywords — Quantum Circuit, quantum gate, Nearest Neighbour(NN), SWAP gate, Quantum Cost(QC).

I. INTRODUCTION

Quantum computing has been found to be more beneficial compared to classical computing over a number of problems. As a result, quantum computing has intensely been considered among the research community. Researchers have been engaged towards the development of quantum computers [1] by using the principles of quantum mechanics. In this regard, designing of quantum algorithms to run on a quantum computer has turned out to be significant.

In quantum computing, quantum algorithms process quantum information by means of quantum circuits, which is a collection of elementary quantum gates. Quantum information is represented in the form of qubits than as bits used in classical computing. Qubits can be used to develop quantum computing machines through various technologies such as ion trap technology [2], nuclear magnetic resonance [3], quantum dots [4] and superconducting qubits [5]. It has been found that the realization of these quantum technologies is subjected to various constraints, noise and drawbacks that need to be addressed otherwise computational errors might arise. Based upon experiments, it has been observed that the nearest neighbor constraint between any two operating qubits is considered as one of the essential requirements to minimize the computational errors. In other words, it indicates that only the interactions between the adjacent qubits are considered as the necessary criteria for the practical realization of quantum circuits. The demand for meeting the nearest neighbor criteria of the interacting qubits has been asserted by the restrictions of J-coupling force needed to enable multi-qubit operations and can only be attained by physically placing the qubits adjacent to each other.

To fix this criterion, the concept of SWAP gates has been introduced. Basically, this gate changes the position of any two qubits rather than its value to satisfy the nearest neighbor constraint. But on the contrary, this addition of SWAP gates in turn increases overhead of the corresponding circuit in terms of both cost and time. In way, incorporation of SWAP gates enhance the quantum cost of the circuit as each SWAP operation can be represented by a group of three CNOT gates (see Fig.1) which in turn increases the number of levels (depth) of the given circuit. This in turn affects the computational time and energy dissipation of the given quantum circuit. Therefore, minimum usage of SWAP gates is considered to be an important objective that needs to be addressed in order to reduce both the execution time and energy dissipation as the number of circuit levels (depth) will be reduced. Albeit exact approaches provide better solutions, but there is no specific optimal solution for very large circuits yet. In recent time several contributory works have been reported where researchers have mainly worked towards reduction of SWAP counts in circuits by adhering different approaches. Here we are stating some of those.

Optimization of NN-compliant circuit by changing the original position of the qubits arranged in linear fashion into a different arrangement has been discussed in [11]. In [12], the number of SWAP gate count has been minimized by arranging the qubits positions dynamically. Optimal solution in terms of number of SWAP gates has been examined in the work [13]. The work of [6, 14] has shown an exact search technique to determine an optimal arrangement of qubits to design a nearest neighbor architecture. Taking a step ahead, to make a trade off in between circuit execution time and SWAP cost a decision based solution for nearest neighbor circuit has been introduced in [22].

As the maximum number of adjacent neighbors of a qubit in one dimension is restricted to only two neighbors and to maximize this parameter, researchers have switched to next higher dimensions - 2D, where a qubit can have a maximum of four adjacent neighbors whereas the qubits located at the periphery can have only two neighbors. In recent time, several

works on 2D circuit design are being reported and here we mentioning few of them.

In one of the work [15], an effective NN-compliant circuit has been designed to reduce overhead in 1D architecture by mapping the problem in 2D architecture through mixed integer programming approach. In [16], the authors used an exact approach to obtain an optimal solution in terms of SWAP cost but it was not suitable for large benchmarks due to exorbitant computational cost. In [17], SWAP cost has been reduced considerable by placing the qubits in appropriate positions. In [18], a heuristic approach has been used to minimize the SWAP cost through proper grid selection followed by an intelligent qubit placement policy where the qubits have been placed upon their number of interactions before adding SWAP gates. To get higher level of optimization, another heuristic approach based on look-ahead strategy has been presented in [19] where substantial improvement in SWAP cost is reported. Aiming to get more improved designs, researchers have tried 3D configurations [20] but it is noticed that though, the number of interacting neighbors increases in 3D but at the same time it becomes very difficult to control the qubits in 3D. In our work we consider the representation in 2D organization only and propose a heuristic qubit placement strategy to convert a quantum circuit to NN-complaint designs with lesser usage of SWAP gates. Our approach provides substantial improvements compared to earlier existing works.

The rest of the paper is organized as follows. Section II describes the elementary quantum gates and nearest neighbor constraints. Detailed descriptions on our proposed approach are stated in Section III. Experimental results and comparative analysis with related works are given in Section IV. Finally, we conclude the paper in section V.

II. BACKGROUND

In quantum computing, qubits represent the basic unit of information on which elementary quantum gates operates. Like bits, a qubit can exist in one of the basis states represented as $|0\rangle$ and $|1\rangle$, similar to 1 and 0 in classical computing. Apart from the basis states, a qubit can also exist in other states expressed as the linear combination of the basis states $|0\rangle$ and $|1\rangle$ represented by the state vector $|\Psi\rangle$ as:

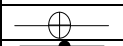
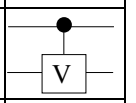
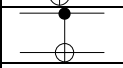
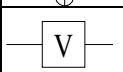
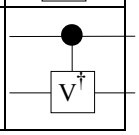
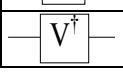
$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where α and β are complex numbers that represents the probability amplitudes of the basis states and subjected to the constraint $\alpha^2 + \beta^2 = 1$.

Definition1. *Quantum gates perform operations on qubits and when a cluster of such gates are projected over a set of circuit lines then a quantum circuit is formed.*

NOT, CNOT, V/V+, W/W+ are well known quantum gates that form the quantum gate libraries (e.g. NCV, NCVW and NCV|v_1> [7-10]) and these gate libraries helps to implement specific functions into gate level descriptions. Some well-known quantum gates and their schematic representations are summarized in Table 1.

Table 1: Some well known elementary quantum gates and their schematic representations

Gates	Notation	Gates	Notation
NOT		Control led -V	
CNOT			
V		Control led -V†	
V†			

Although quantum gates can express any function into respective circuital form, there are certain limitations for practical realization of such type of circuits. One such major criterion is to restrict the quantum gates to operate only on qubits located adjacent to each other. This adjacency feature is required to mitigate the errors induced in the circuit from various sources of noise. For this purpose, SWAP gates are being introduced before a quantum gate that fails to meet this adjacency constraint by exchanging the positions of the respective qubits till the condition has been met. This phenomenon is known as Nearest Neighbor property, which can be defined as follows:

Definition2: *Nearest Neighbor Cost (NNC) is the interaction distance between the qubits of any two-qubits quantum gate (G) and can be expressed as* $NNC_G = |c - t| - 1$*, where c is the control line and t is target line of the gate.*

The NNC pertaining to a circuit (NNC_{CKT}) is the cumulative value of NNCs contributed by the individual gates i.e. $NNC_{CKT} = \sum_G NNC_G$, NNC_G is NNC of each two-qubit gate in the circuit.

For example, the standard decomposition of a Toffoli gate has NNC = 1 due to the presence of a single non-adjacent elementary gate in its gate level description.

Though, several researches are going on efficient transformation of quantum circuit to NN compliant designs, but the most common way of converting a quantum circuit to a NN compliant one is by inserting SWAP gates (the pictorial representation of SWAP gate is shown in Fig. 1). Basically, such SWAP gates are used to bring the interacting qubits of any gate adjacent.

Example1: *For example, the design of Fig. 2(a) does not satisfy the nearest neighbor constraint as the qubits in the first, second, fourth and fifth gates are non-adjacent to each other. So, SWAP gates have to be inserted to meet this constraint. The transformation of the Fig. 2(a) to a NN complaint design is achieved by inserting 14 SWAP gates in the structure and the final 1D NN-complaint organization is shown in Fig. 2(b).*

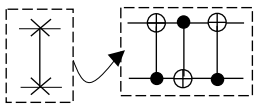


Fig. 1: Design of SWAP gate

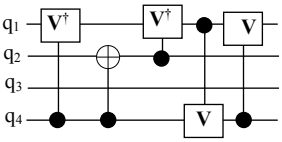


Fig. 2(a): Input quantum circuit having NNC=7

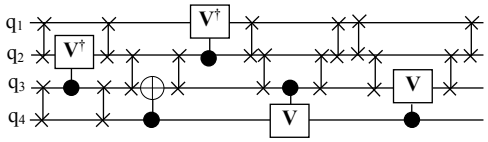


Fig. 2(b): Equivalent NN-compliant design of Fig. 2(a)

Though we have converted the circuit to a 1-D based NN compliant form, but it is also possible to represent the same circuit in 2-D format as well to get improved representation. Now, here we are stating how this 2D design can be made.

2D Quantum Circuit: To map a quantum circuit in 2D configuration, primarily a grid structured planar graph (V, E) has been used where a node v represents a qubit and an undirected edge $(u,v) \in E$ represents the interaction between the qubits u and v, where $\{u,v\} \in V$. Here, each node can have a degree of up to four. Basically, a quantum circuit may be represented in several 2D configurations based upon the chosen grid structures. For instance, we consider a 1D circuit with 5 qubits. In this case, these 5 qubits are always arranged next to each other while these same qubits when arranged in a 2D quantum circuit, may have several configurations such as a 3x2 or 2x3 or 3x3 grid as shown in Fig. 3. Such different arrangements affect the interaction distance between the corresponding qubits.

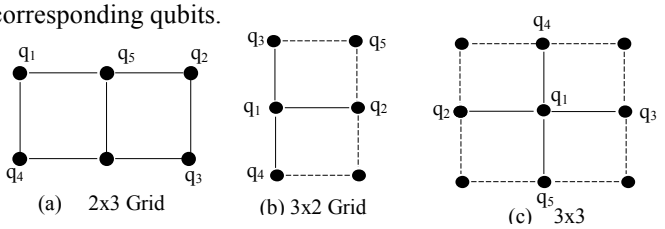


Fig. 3: Possible grid configuration for qubit positions in 2D

As the 1D representation of Fig. 2(b), for the 2D organization a 2x2 grid is now selected and gates are placed over the circuit lines as depicted in Fig. 4(a). The NN-complaint design in 2D form is obtained and the final circuit after SWAP insertion is shown in Fig. 4(b).

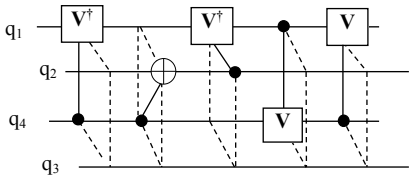


Fig. 4(a): 2D representation of Fig. 2(a)

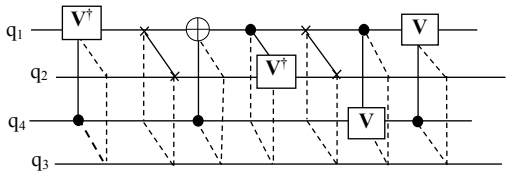


Fig. 4(b): 2D NN-compliant design of Fig. 4(a)

III. PROPOSED APPROACH

Here, we are proposing an improved synthesis scheme for converting quantum circuits to its equivalent 2D NN design. This technique not only transforms the input quantum circuits into nearest neighbor based design but also limits the SWAP utilization to contain the circuit overhead. In the whole transformation process, we have taken help of some heuristics in decision making points and have derived an intelligent qubit placement policy for better representation.

The entire scheme is performed in three phases – (i) qubit selection (ii) qubit placement and (iii) SWAP gate insertion. For an ease of understanding, all these phases are explained here with an example and the circuit of Fig. 5(a) has been considered as standard input circuit over which all the operations are to be performed.

Phase1: Qubit Selection Policy

This stage is tasked with proper arrangement of qubits in a grid structure. But to make this preference enable it needs some preference metrics. For this purpose, we build two information tables (*time interaction* and *time costing*) from the gate details of the input circuit.

In *time interaction* table, we compute total interaction time of each qubit. The entries of this table comprise the time instants in which the corresponding qubits interacts and the total interaction time which is computed by summing all the time instants of interactions for every qubit in the circuit. The *time interaction* metric for Fig. 5(a) is shown in Table 2.

Next we compute, the qubit *time costing* table (shown in Table 3) which contains the total costing time for each qubit and this metric is derived from our earlier computed table, qubit *time interaction*.

The total costing time of qubit *time costing* table is estimated for each qubit after extracting the time instants from the timestamps of the corresponding qubit interaction in which it is not adjacent with its other interacting qubit of any 2-qubit gate in the given circuit.

Combining the information provided in this two tables, we derive one more table, *qubit preference* table (given in Table 4) in which we compute the preference index by calculating the ratio between total costing time and total interaction time in a separate column for all the qubits of a given circuit.

After obtaining the preference index for all the qubits in the given circuit, we sort all the qubits based upon this index values in a descending order (see Table 5) with the qubit having the highest index value is placed at the top of the table. The preference index values obtained for each qubit in *qubit preference table* designates the priority values of the corresponding qubits for placement selection in the grid structure.

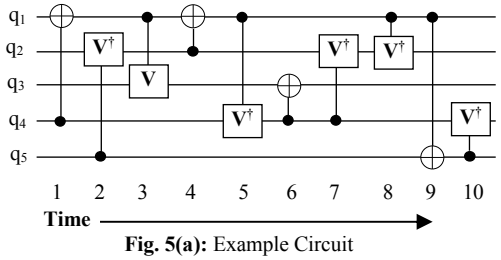


Fig. 5(a): Example Circuit

Table 2: Qubit time interaction table

Qubits	Time instants	Total interaction
q_1	1,3,4,5,8,9	30
q_2	2,4,7,8	21
q_3	3,6	9
q_4	1,5,6,7,10	29
q_5	2,9,10	21

Table 3: Qubit time costing table

Qubits	Time instants	Total interaction	Time instants	Total costing
q_1	1,3,4,5,8,9	30	1,3,5,9	18
q_2	2,4,7,8	21	2,7	9
q_3	3,6	9	3	3
q_4	1,5,6,7,10	29	1,5,7	13
q_5	2,9,10	21	2,9	11

Table 4: Qubit preference table

Qubits	Total interaction	Total costing	Preference index
q_1	30	18	18/30=0.6
q_2	21	9	9/21=0.42
q_3	9	3	3/9=0.33
q_4	29	13	13/29=0.44
q_5	21	11	11/21=0.52

Table 5: Sorted qubit preference table

Qubits	Total interaction	Total costing	Preference index
q_1	30	18	0.6
q_5	21	11	0.523
q_4	29	13	0.448
q_2	21	9	0.428
q_3	9	3	3/9=0.33

Phase2: Qubit Placement Policy

After computing the qubit preference table in the previous execution process (refer to Table 5), we now hereby run the qubit placement algorithm by arranging the qubits chosen in the order from Table 5 in a grid structure (given in algorithm 1).

Algorithm 1: Qubit placement strategy

Input: Qubit preference table (PT)
Output: Qubit placement in 2D grid structure (GS)

```
begin
  for ((qk ∈ PT) and (PT≠null))  do
    if k = 1 then
      Place qk in the centre of 2D grid GS (mid_row,mid_column);
    else
      for k =2 to N do                          //N is the number of qubits
        Find a vacant location GS (row, column) adjacent to maximum
        number of empty cells;
        if ((Num_of_(GS (row, column)) = = 1)  then
          Place qk in location GS (row, column);
        else
          Retrieve location GS (row_{k-1}, column_{k-1}) of last qubit q_{k-1};
          Place qk in one of the vacant locations from GS (row_{k-1} ± a,
                                                  column_{k-1} ± b); where a, b = 1, 2
        end if
      end for
    end if
  end for
  return GS;
end
```

According to the steps mentioned in that algorithm, initially, we select the qubit having the highest preference index from the qubit preference table (PT) and place it in the centre of the 2D grid. Then we start finding an empty location in that grid which is adjacent to maximum number of empty cells and place the next qubit from PT table.

If there occur multiple occurrences of such a type of grid location then we resolve the conflict by selecting the qubits from PT and then placing it in these locations in the order left, up, right, down of the last placed qubit as per the availability of space until such a distinct location is being sought. In this way, we populate the 2D grid by placing the remaining qubits from the preference table (PT).

We employ the same process over our previously computed preference table to find a suitable grid configuration. As the qubit q_1 is having the highest preference in PT (Table 5), so we place the selected qubit in the centre of the grid (3×3) (see Fig. 5(b)). Now, we search for a location with maximum number of adjacent empty cells in the grid. Since, there are eight such possible vacant locations exist with two empty cells around, so we place the qubits from PT around the left, up, right, down of the last placed qubit q_1 till a distinct location with empty cells around it is found. Thus, we place the next qubit q_5 to the left of the qubit q_1 as stated in our algorithm 1(see Fig. 5(c)). Now, the rest of the qubits q_4, q_2, q_3 are selected from PT and positioned them at the up, right and down of q_1 respectively. The final grid configuration is given in Fig. 5(d).

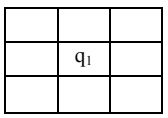


Fig. 5(b): q_1 inserted in 3×3grid

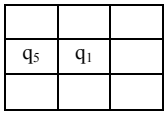


Fig. 5(c): q_5 inserted to the left of q_1

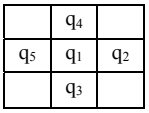


Fig. 5(d): Qubits q_4 , q_2, q_3 placed around q_1

Phase3: SWAP gate insertion

Till now, all the qubits have been positioned in the grid. In this phase, we place all the gates over this grid in such a way that we insert a SWAP gate if we find any gate whose interacting qubits are non-adjacent. In this process, we simply exchange the position of qubits and make them adjacent.

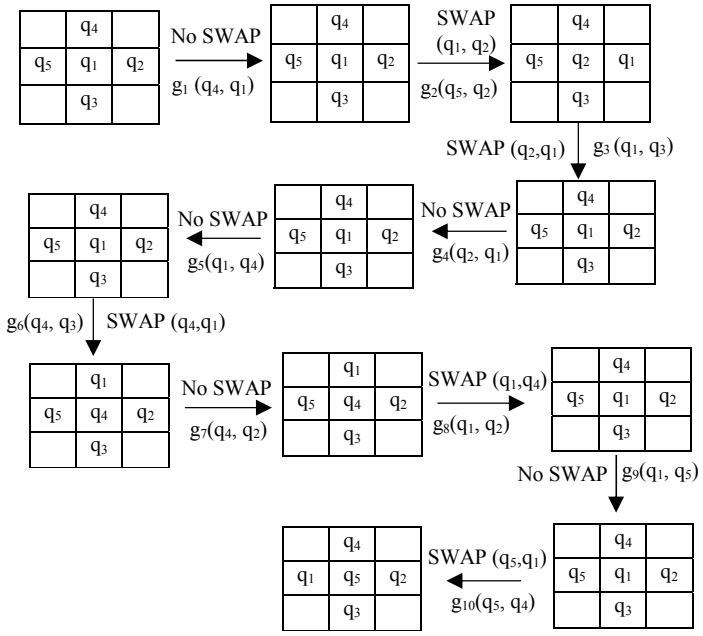


Fig. 6: Steps of SWAP gate insertion

For our input circuit, we already have obtained the desired grid (in Fig. 5(d)) and now we have placed the qubits accordingly as the mentioned strategy. The placement of qubits and insertion of SWAP gates in appropriate places are given in Fig. 6, where it can be found that a total of five SWAP gates have been utilized in the entire transformation process. For better understanding, here we are producing one more example.

Example2: The transformation of benchmark circuit (4gt11_84) to its NN equivalent form is shown in this example. The intermediate steps and computed metrics are also given in Table 6-10.

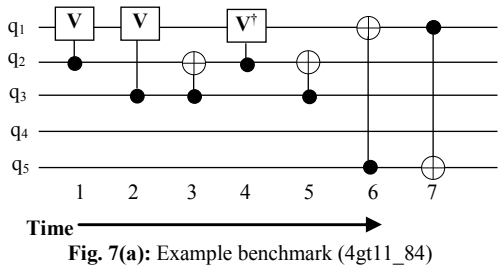


Fig. 7(a): Example benchmark (4gt11_84)

Table 6: Qubit time interaction table

Qubits	Time instants	Total interaction
q_1	1,2,4,6,7	20
q_2	1,3,4,5	13
q_3	2,3,5	10
q_4	-	-
q_5	6,7	13

Table 7: Qubit time costing table

Qubits	Time instants	Total interaction	Time instants	Total costing
q_1	1,2,4,6,7	20	2,6,7	15
q_2	1,3,4,5	13	0	0
q_3	2,3,5	10	0	0
q_4	-	-	-	-
q_5	6,7	13	6,7	13

Table 8: Qubit preference table

Qubits	Total interaction	Total costing	Preference index
q_1	20	15	15/20=0.75
q_2	13	0	0/13=0
q_3	10	0	0/10=0
q_4	-	-	-
q_5	13	13	13/13=1

Table 9: Sorted qubit preference table

Qubits	Total interaction	Total costing	Preference index
q_5	13	13	13/13=1
q_1	20	15	15/20=0.75
q_2	13	0	0/13=0
q_3	10	0	0/10=0
q_4	-	-	-

Table 10: Qubits placed in grid (2×3)

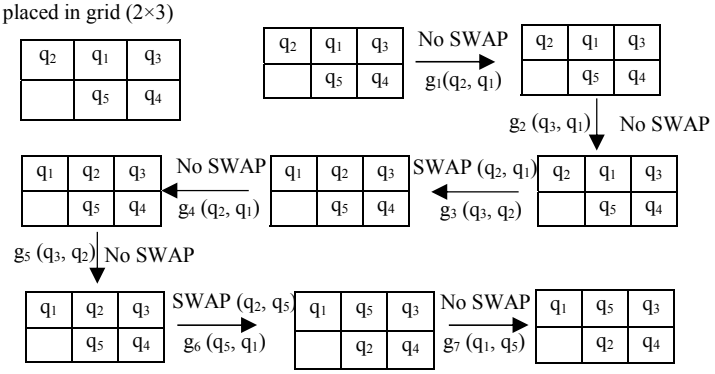


Fig. 7(b): Steps of SWAP gate insertion

IV. Experimental Results

The proposed approach has been implemented in C and executed on Intel i5 machine with 3.30 GHz clock and 4GB RAM. The experimental evaluations have been carried out over different benchmark suites from [21]. Two result tables have been produced, where the first table (Table-11) contains results from small and medium size benchmarks and the second table (Table-12) summarizes the experimental data for large size benchmarks.

For each of the benchmarks two cost parameters – (quantum cost and SWAP count) have been evaluated and the obtained metrics have been compared with some well-known 1D and 2D works. The first result table contains the comparison with [15], [18], and the best results from the work of [15] have been considered. A steady improvement in cost parameters can be seen in the obtained results. 28.71% average improvement over 1D works and 10.18% average improvement over 2D works have been attained. Our qubit mapping policy results in an improvement of about 37.73% and 35.71% respectively in the best case over the 2D works [15] and [18], while an improvement of 67.56% has been obtained in the best case over the 1D work [15]. Experimental analysis suggests that our approach provides promising results in terms of SWAP gate reduction for all but few benchmarks for which our mapping strategy has not provided better results relative to the previous works.

V. Conclusion

This work presents a heuristic based qubit placement strategy for transformation of quantum circuits to NN-complaint design. The entire design processes has involved three stages - qubit selection then placement and finally SWAP gate insertion. We have tested the developed approach over a large spectrum of benchmarks and improvements have been registered. A substantial reduction in SWAP gate usage has been achieved. The obtained results have been compared with some of the best results available in 1D and 2D representations. Depending on our experimental evaluations, we examined that our approach has attained an average reductions of about 35.32%, 22.10% in terms of SWAP cost and quantum cost against 1D configuration while 17.09%, 3.28% reductions against 2D configuration over the same cost parameters.

As we have highly relied on the applied heuristic for initial qubit mapping process, so it does not provide optimal solution. Further enhancement over this work will be investigated in future by finding more appropriate heuristics.

VI. References

[1] T. Simonite. IBM shows off a quantum computing chip. http://www.technologyreview.com/news/537041/ibm-shows-off-a-quantum-computing-chip/, 2015.

[2] D. Kielpinski, C. Monroe, and D. J. Wineland. “Architecture for a largescale ion-trap quantum computer”, *Nature*, 417(6890):709–711, 2002.

[3] B. Criger, G. Passante, D. Park, and R. Laflamme. “Recent advances in nuclear magnetic resonance quantum information processing”. *Philosophical Transactions of the Royal Society of London A: Mathematical,Physical and Engineering Sciences*, 370(1976):4620–4635, 2012.

[4] J. Taylor, J. Petta, A. Johnson, A. Yacoby, C. Marcus, and M. Lukin, “Relaxation, dephasing, and quantum control of electron spins in double quantum dots”. *Physical Review B*, 76(3):035315, 2007.

[5] A. Blais, J. Gambetta, A. Wallraff, D. Schuster, S. Girvin, M. Devoret, and R. Schoelkopf. “Quantum information processing with circuit quantum electrodynamics”. *Physical Review A*, 75(3):032329, 2007.

[6] A. Zulehner, S. Gasser, R. Wille. “Exact Global Reordering for Nearest Neighbor Quantum Circuits Using A* . In *International Conference on Reversible Computation* (pp. 185-201). Springer, Cham. 2017, July

[7] A. Barenco, C. H. Bennet, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H Weinfurter, “Elementary gates for quantum computation,” *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, Nov 1995.

[8] Z. Sasnian and D. Miller, “Transforming MCT Circuits to NCVW Circuits,” *Springer's Lecture Notes in Computer Science.*, pages 77-88, (2011).

[9] D. Miller, R. Wille, and Z. Sasanian, “Elementary quantum gate realizations for multiple-control Toffolli gates,” *In Proc. Int'l Symp. on Multiple-valued Logic*,pages pp. 217-222, (2011).

[10] Z. Sasanian, R. Wille, and D. M. Miller, "Realizing reversible circuits using a new class of quantum gates," *In Proc. Design Automation Conf.*, pp. 36-41, (2012).

[11] Y. Hirata, M. Nakanishi, S. Yamashita, Y. Nakashima. "An efficient conversion of quantum circuits to a linear nearest neighbor architecture." *Quant. Inf. Comput.*, 11(1–2):0142–0166, 2011.

[12] M. Saeedi, R. Wille, R. Drechsler. "Synthesis of quantum circuits for linear nearest neighbor architectures." *Quant. Inf. Proc*., 10(3):355–377, 2011.

[13] A. Shafaei, M. Saeedi, and M. Pedram. Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures. *Design Autom. Conf.*, 2013.

[14] R. Wille, A. Lye, and R. Drechsler, "Exact reordering of circuit lines for nearest neighbor quantum architectures," *IEEE Trans. on CAD*, vol. 33, no. 12, pp. 1818–1831, Dec 2014.

[15] A. Shafaei, M. Saeedi, and M. Pedram, "Qubit placement to minimize communication overhead in 2d quantum architectures," in *Proc. ASP Design Autom. Conf.*, Jan 2014, pp. 495–500.

[16] A. Lye, R. Wille, and R. Drechsler, "Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits," *in Proc. ASP Design Autom. Conf.*, Jan 2015, pp. 178–183.

[17] M. G. Alfailakawi, I. Ahmad, and S. Hamdan, "Harmony-search algorithm for 2D nearest neighbor quantum circuits realization," *Expert Systems with Applications*, vol. 61, pp. 16–27, May 2016.

[18] R. R. Shrivastwa, K. Datta, and I. Sengupta, "Fast qubit placement in 2D architecture using nearest neighbor realization," *in Proc. Int'l Symp.on Nanoelectronic and Information Systems*, Dec 2015, pp. 95–100.

[19] R. Wille, O. Keszocze, M. Walter, P. Rohrs, A. Chattopadhyay, and R. Drechsler, "Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits," in *Proc. ASP Design Autom. Conf.*, Jan 2016, pp. 292–297.

[20] A. Kole, K. Datta, I. Sengupta, "A New Heuristic for N-Dimensional Nearest Neighbour Realization of a Quantum Circuit". *In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 12 (2017), doi 10.1109/TCAD.2017.2693284.

[21] Revlib: An online resource for reversible functions and reversible circuits. URL: http://www.revlib.org/.

[22] R. Wille, N. Quetschlich, Y. Inoue, N. Yasuda, and S. Minato. "Using π-DDs for Nearest Neighbor Optimization of Quantum Circuits" *In Conference on Reversible Computation*, pages 181-196, 2016.

Table12: Results of large size benchmarks

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

Table11: Comparison over 2D results with small and medium size benchmark circuits

Benchmarks	No. of qubits	Gate count	Results in 1D from [15]		Results in 2D from[15]			Results from [18]			Proposed work results		
			SWAP Count	Quantum Cost	Grid Size	SWAP Count	Quantum Cost	Grid Size	SWAP Count	Quantum Cost	Grid Size	SWAP Count	Quantum Cost
3_17_13	3	14	4	17	2×2	6	20	2×2	3	17	2×2	5	19
4_49_17	4	32	12	45	2×2	13	45	-	-	-	2×2	10	42
aj-e11_165	4	60	36	96	2×3	24	84	3×2	22	82	2×3	18	78
decod24-v3_46	4	9	3	12	3×2	3	12	-	-	-	2×2	2	11
hwb4_52	4	23	10	33	2×2	9	32	2×2	9	32	2×2	7	30
rd32-v0_67	4	8	2	10	2×3	2	10	-	-	-	2×3	2	10
4gt11_84	5	7	1	8	2×3	2	9	2×3	2	9	2×3	2	9
4gt10-v1_81	5	36	20	56	3×2	16	52	3×2	15	51	2×3	14	50
4gt12-v1_89	5	53	35	88	3×2	19	72	2×4	18	71	3×2	20	73
4mod5-v1_23	5	24	9	33	2×3	11	35	3×2	7	31	2×3	8	32
4gt5_75	5	22	12	34	3×3	8	30	2×5	10	32	3×3	9	31
4gt4-v0_80	5	44	34	78	2×3	17	61	4×4	15	59	2×3	16	60
4mod7-v0_95	5	40	21	61	3×3	13	53	2×5	14	54	3×3	10	50
alu-v4_36	5	32	18	50	2×3	10	42	2×5	11	43	2×3	11	43
hwb5_55	5	109	63	172	3×2	45	154	2×7	49	158	3×2	38	147
QFT5	5	10	6	16	3×2	5	15	4×2	5	15	3×2	5	15
hwb6_58	6	146	118	264	2×3	79	225	2×3	76	222	2×3	63	209
mod5adder_128	6	87	51	138	3×2	41	128	2×3	36	123	3×2	28	115
mod8-10_177	6	109	72	181	3×3	45	154	4×3	43	152	3×3	39	148
QFT6	6	15	12	27	2×3	6	21	3×2	7	22	2×3	5	20
rd53_135	7	78	66	144	5×2	39	117	2×7	40	118	3×3	29	107
ham7_104	7	87	68	155	3×3	48	135	2×7	45	132	3×3	38	125
QFT7	7	21	26	47	2×4	18	39	6×2	14	35	2×4	14	35
QFT8	8	28	33	61	4×2	18	46	4×2	23	51	4×2	18	46
QFT9	9	36	54	90	3×3	34	70	5×2	36	72	3×3	24	60
rd73_140	10	76	56	132	4×3	37	113	3×6	43	119	4×3	28	104
sys6-v0_144	10	62	59	121	4×4	31	93	-	-	-	4×3	30	92
QFT10	10	45	70	115	5×3	53	98	4×3	51	96	4×3	33	78
rd84_142	15	112	148	260	5×3	54	166	4×5	62	174	5×3	48	160
cnt3-5_180	16	125	127	252	3×6	69	194	4×4	84	209	3×6	54	179
Total			**971**	**2796**		**775**	**2325**		**740**	**2179**		**628**	**2178**