

Exploiting the Benefits of Clean Ancilla Based Toffoli Gate Decomposition Across Architectures

Abhoy Kole¹, Kamalika Datta^{1,3}, Philipp Niemann¹, Indranil Sengupta², and Rolf Drechsler^{1,3}

¹ Cyber-Physical Systems, DFKI GmbH, Bremen, Germany {abhoy.kole, Philipp.Niemann}@dfki.de

² Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India isg@iitkgp.ac.in

³ Institute of Computer Science, University of Bremen, Bremen, Germany {kdatta,drechsler}@uni-bremen.de

Abstract. Elementary gate decomposition of larger Toffoli operations is often carried out using additional qubits (ancilla). The number of gates and the circuit depth vary in such transformation depending on the type of ancilla used (clean or dirty). The present *Noisy Intermediate Scale Quantum* (NISQ) devices have limited number of coherent qubits with faulty native operation support. Superconducting devices also have coupling restrictions or *Nearest-Neighbor* (NN) constraints, which require additional gates to map the transformed netlist for execution. While the mapping overhead is correlated with the number of 2-qubit gates and involved qubits, the fidelity of execution is inversely proportional to the number of gates and circuit depth. There is a tradeoff in devising the transpilation (i.e. low-level transformation and mapping) approach — dirty ancilla demands less qubits and overhead at the expense of more gates and depth as compared to clean ancilla, which involves less gates and depth at the expense of more qubits and overhead. This paper analyzes the disparity in gates, depth and qubits between: (i) the low-level transformation approaches without considering device coupling information, and (ii) the mapping schemes based on netlist transformation using a specific type of ancilla. We analyze the benefits of using NN-constraints at the transformation stage, and the impact of distributing clean ancilla across architectures. We have carried out experiments on IBM Q20 and Hexagonal Q20 architectures, which show improvements of 17% and 13% respectively in terms of number of gates.

Keywords: Quantum circuits · Architecture-aware decomposition · Qubit mapping · Clean and dirty ancilla.

1 Introduction

We have witnessed rapid developments in quantum computers over the last few years with several prominent physical implementations (viz. by IBM, Google, DWave and many others). These implementations are often referred to as *Noisy*

Intermediate Scale Quantum (NISQ) era devices due to their faulty native operations within a small number of limited coherent qubits. Additionally, the coupling restrictions or *Nearest-Neighbor* (NN) constraints between qubits, built using technologies like superconducting elements, require additional gates to perform 2-qubit operations on any pair of uncoupled qubits. While the number of operations (gates) and latency (circuit depth) of a computation pose a threat to its reliability, the overhead further leads to the accumulation of noise.

The transpilation of quantum circuits for a NN-architecture is typically a two-step process. Initially, a given netlist realized using multi-qubit operations like *Multiple-Control Toffoli* (MCT) gates is transformed into a low-level description using 1- and 2-qubit elementary gates from a specific library, e.g. the Clifford+T. The gates are then mapped satisfying the NN-constraints of the target architecture. For such low-level description of MCT operations, the use of additional qubits (ancilla) is very effective in reducing the gates and depth of the netlist [7]. While the use of *dirty ancilla* reduces the qubit requirement by reusing circuit qubits, the gates and depth can be further minimized when qubits not used in the circuit are used as *clean ancilla* in the transformation [11]. During the final stage of transpilation, NN-compliance is achieved by introducing more gates to satisfy the NN-constraints.

There exist several works for transforming MCT operations into elementary gate netlists [4, 14] with varying gates, depth and qubits. The mapping overhead (i.e. number of 2-qubit gates and required qubits) is inversely related to the computational reliability. Most of the mapping methods have used generated netlists obtained from standard transformations without considering the device constraints [9, 18]. For quantum devices with NN-constraints [1, 15], it is important to consider architectural information at the transformation stage to reduce gates and depth of the transpiled netlist. In a recent work Baker et al. [5] have shown decomposition of Toffoli gate with arbitrary number of clean and dirty ancilla qubits. In another related work, Balauca et al. [6] have proposed efficient construction of multi-control quantum gates. However, no specific analyses considering the architectural information to map these decomposed netlists have been carried out so far.

One of the recent works [12] combine *Swap* gates and *Remote-CNOT* templates to generate a database of MCT operations transpiled using dirty ancilla for the IBM Q20 architecture. With the gradual increase in the number of qubits, more number of qubits can be used for transpilation. To this end, in this paper we exploit clean ancilla based transpilation of MCT gate netlists with NN-compliance. We show that the use of clean ancilla outperforms the approach that uses dirty ancilla, and analyze the impact of clean ancilla distribution on two architectures, viz. 20-qubit IBM Q20 and Hexagonal Q20. Experimental results confirm that clean ancilla based approach is beneficial in reducing the gate overhead. Overall improvements of up to 17% and 13% respectively are obtained for IBM Q20 and Hexagonal Q20 architectures.

Rest of the paper is organized as follows. Section 2 provides the necessary background and motivation for the work. Section 3 presents the proposed

database-driven architecture-aware decomposition approach. Section 4 presents the experimental results, and finally the conclusion is provided in section 5.

2 Background and Motivation

In this section we discuss about quantum gates and architectures, and some of the cost metrics used for decomposition.

2.1 Quantum Gates and Quantum Circuits

In quantum computing, the basic unit of information is quantum bits or *qubits*. The state of a qubit can be represented as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex amplitudes such that $|\alpha|^2 + |\beta|^2 = 1$. A qubit can be in a state of superposition, and also a set of qubits can exist in states of entanglement. Quantum gates operate on qubits and change their states; typically, we have 1- and 2-qubit primitive gate operations. A quantum circuit consists of a sequence of quantum gates, where each m -qubit quantum gate can be represented by a $2^m \times 2^m$ unitary matrix. Also, every quantum gate operation is inherently reversible.

A Toffoli gate $T(\{c_1, c_2\}; t)$ is a 3-input reversible gate that has two controls c_1 and c_2 and a target t . If the values of c_1 and c_2 are both 1, then the target t changes to $c_1c_2 \oplus t$. Fig. 1 shows a Toffoli gate and its realization using Clifford+T gate library. Another gate library that has been used by many researchers is known as NCV [7].

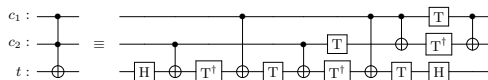


Fig. 1: Toffoli gate and its Clifford+T realization.

2.2 Quantum Architecture

Several implementations of quantum computers have emerged in the last five years [2, 3, 8, 10, 15]. Most of these devices are built using superconducting qubits. Recently photonic technologies have also been explored to realize qubits [15]. Fig. 2 shows two different NN architectures each comprising of 20 qubits and having maximum degree of qubit coupling of six.

For mapping quantum circuits, additional gates are required to comply with the architectural limitations or NN-constraints [10]. Typically, Swap gates [17]

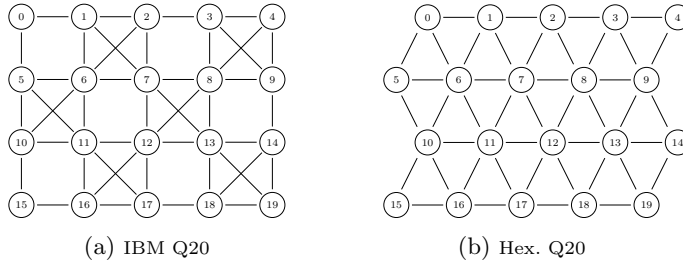


Fig. 2: Qubit coupling layout for two 20-qubit architectures.

or remote-CNOT templates [13] are inserted in the netlist for compliance. As the gate operations are noisy, the inclusion of extra gates accumulate more errors in the final results. Hence reducing the number of gates has a direct bearing in controlling the error. Most of the existing works in the literature consider netlists consisting of 1- and 2-qubit gates that are generated using existing transformation methods. However, by exploiting information about the architecture (viz. number of qubits, inter-qubit coupling, error statistics, etc.), we can suitably select the ancilla lines so as to minimize the gate overhead. One recent work [12] has considered the IBM Q20 architecture to generate a complete database for MCT gates using dirty ancilla; however, the possibility of clean ancilla usage is unexplored. In this paper we explore the benefits of using clean ancilla during gate decomposition, and show how MCT gate networks can be mapped to various architectures efficiently.

2.3 Cost Metric and WCOST

To execute quantum gates with NN-compliance, two broad approaches exist, both of which incur the use of additional CNOT gates in the netlist. The first approach tries to bring the states of a pair of interacting qubits to adjacent locations by adding Swap gates [17]. As an alternative, a 2-qubit gate operation on non-adjacent qubits can be carried out as a cascade of NN-compliant CNOT operations, called remote-CNOT template [13]. In the latter case, the number of additional gates required is $4d$, where d is the distance between the pair of physical qubits. The cost is estimated as the number of additional CNOT gates.

For the netlist of Fig. 1, the *Qubit Interaction Graph* (QIG) is shown in Fig. 3(a). The edge weights represent the number of 2-qubit operations between qubit pairs. We calculate the total cost (WCOST) for mapping the netlist to IBM Q20 and Hexagonal Q20 architectures. With respect to the QIG and a layout mapping, $\pi : q_i \rightarrow Q_k$, let E denote the set of edges, w_{ij} the edge weight between qubits q_i and q_j , and d_{ij} the number of qubits in the shortest coupling path between $\pi(q_i)$ and $\pi(q_j)$. The total cost can be estimated as:

$$WCOST = \sum_{(i,j) \in E} C_{ij} \text{ where } C_{ij} = \begin{cases} w_{ij} & \text{if } d_{ij} = 0 \\ 4d_{ij}w_{ij} & \text{otherwise.} \end{cases} \quad (1)$$

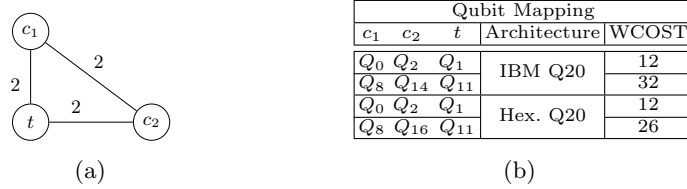


Fig. 3: (a) Interaction graph of a Toffoli gate $T(\{c_1, c_2\}; t)$, and (b) corresponding WCOST for mapping into IBM Q20 and Hexagonal Q20 architectures.

Fig. 3(b) shows the WCOST for four different mappings for the two architectures.

3 Proposed Method

In this work we show how clean ancilla is beneficial as compared to dirty ancilla for MCT gate decomposition. We first analyze ancilla usage in realizing Toffoli gates using primitive quantum gates. The mapping overhead on two specific architectures, viz. IBM Q20 and 20-qubit regular hexagonal, are presented next. Finally, the effects of clean ancilla distribution on given qubit layout are examined.

3.1 Realization of MCT Netlist

The transformation of an $(m + 1)$ -qubit MCT gate requires $m - 2$ ancilla qubits for $m > 2$. The number of gates in the transformed netlist may vary depending on whether clean or dirty ancilla are used. A $(m + 1)$ -qubit MCT gate (for $m > 2$) requires $4(m - 2)$ 3-qubit Toffoli gates when $m - 2$ dirty ancilla are used [7], and $(2m - 3)$ Toffoli gates when $m - 2$ clean ancilla are used [11].

Example 1. Fig. 4 shows the realization of a 5-qubit MCT gate, $T(\{c_1, \dots, c_4\}, t, \{a_1, a_2\})$ decomposed using 8 and 5 Toffoli gates considering a_1 and a_2 as dirty and clean ancilla qubits, respectively.

The number of gates can be further reduced by performing some optimization. In general, the operation of a Toffoli gate can be realized as a sequence of 3-qubit $C^2(-iZ)$ ($C^2(iZ)$) and CS (CS^\dagger) operations. For such Toffoli gate pairs operating on same set of qubits in presence of an intermediate gate, the component CS (CS^\dagger) gate pair can be removed completely (partially) depending on the intermediate gate operation. This is illustrated by the following example.

Example 2. Fig. 5 shows the netlists comprising of pair of 3-qubit Toffoli gate, $T(\{q_1, q_2\}; q_3)$, with an intermediate CNOT gate, $CX(q_c, q_t)$. The component CS (CS^\dagger) from the netlist can be removed completely, if the control, q_c is q_3

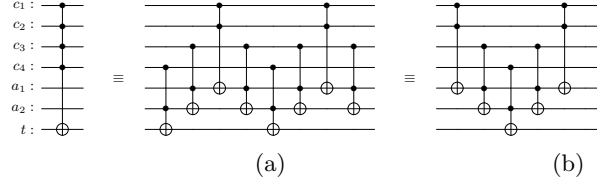


Fig. 4: Decomposition of a 4-controlled MCT gate as a netlist of Toffoli gates using (a) 2 dirty ancilla, (b) 2 clean ancilla.



Fig. 5: A pair of Toffoli gates and an intermediate CNOT gate operating on the (a) target and (b) control qubits of the Toffoli gates.

and target, $q_t \notin \{q_1, q_2\}$ (see Fig. 5a). Similarly, for the netlist shown in Fig. 5b, one CNOT operation from each of the CS (CS^\dagger) realization can be removed, when the q_2 (or q_1) becomes q_t .

Considering such cancellation, the number of CNOT operations to realize a $(m+1)$ -qubit MCT gate (for $m > 2$) gets reduced to $(20m - 42)$ when $m - 2$ dirty ancilla are used. Similarly, for $m - 2$ clean ancilla, the required number of CNOT operations is reduced to $(8m - 10)$. However, it may be noted that the ability to reuse circuit qubit as ancilla leads to a final netlist with fewer qubits when dirty ancilla are used for decomposition instead of clean ancilla.

Lemma 1. *A r -qubit MCT network can be described optimally using 1- and 2-qubit Clifford+T gates and r qubits, when only dirty ancilla qubits are used and the largest MCT gate in the network is of size $(m+1)$ -qubit where $m \leq \frac{r+1}{2}$.*

Proof. In order to describe the operation of an $(m+1)$ -qubit MCT gate using minimum number of Clifford+T gates, we require $m - 2$ dirty ancilla qubits. Thus, for a r -qubit MCT network, the Clifford+T description does not require any additional qubits, when the largest MCT gate in the netlist is of size $(m+1)$ -qubit, and it satisfies the condition, $(m+1) + (m-2) \leq r$ i.e., $m \leq \frac{r+1}{2}$.

The Clifford+T description of an MCT gate netlist requires additional qubits other than circuit when clean ancilla are used.

Lemma 2. *A r -qubit MCT network can be described optimally using $r + m - 2$ qubits and 1- and 2-qubit Clifford+T gates when only clean ancilla are used and the largest MCT gate is of size $(m+1)$ -qubit.*

Proof. In order to describe the operation of the $(m + 1)$ -qubit MCT gate using minimum number of Clifford+T gates, we require $m - 2$ clean ancilla. Thus, for a r -qubit MCT netlist the Clifford+T description requires $(r + m - 2)$ qubits.

To summarize, the use of clean ancilla results in a netlist with fewer gates as compared to that using dirty ancilla. On the other hand, use of dirty ancilla requires fewer qubits. Thus, increasing the number of qubits in re-describing a MCT netlist can reduce the number of gates in the final netlist as illustrated by the following example.

Example 3. Fig. 6 shows two equivalent descriptions of a 5-qubit MCT netlist using clean and dirty ancilla qubits. The use of dirty ancilla requires 5 Toffoli gates with no additional qubits (vide Lemma 1). The use of clean ancilla requires one additional qubit (q_a) and 4 Toffoli gates (vide Lemma 2).

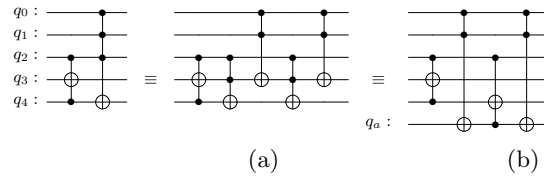


Fig. 6: Decomposition of a MCT netlist using (a) dirty ancilla, (b) clean ancilla.

Clearly, there is a tradeoff in using either clean or dirty ancilla qubits for decomposition. In addition, while mapping the gate operations to some physical architecture, the NN-constraints must be taken into account. This is discussed in the next subsection.

3.2 Architectural Mapping of MCT Netlist

Current NISQ architectures have limited number of qubits, and mapping a r -qubit MCT netlist requires re-describing it in terms of native 1- and 2-qubit gates. Of course, the number of qubits in the final netlist cannot exceed the available number of physical qubits. The number of qubits in the decomposed netlist depends on two factors: (i) size of the largest MCT gate in the netlist, and (ii) the type of ancilla used in decomposition. This limits the size of the MCT netlist that can be mapped to n physical qubits.

Lemma 3. *In an n -qubit target architecture, an n -qubit MCT netlist can be mapped if the largest MCT gate present in the network never occupies more than 50% (or $\lfloor \frac{n+1}{2} \rfloor + 1$) of n qubits, where dirty ancilla are used in the decomposition.*

Proof. Decomposing an $(m + 1)$ -qubit MCT gate requires $m - 2$ ancilla qubits, where $m > 2$. To map the decomposed netlist on an n -qubit architecture, it must

satisfy the inequality: $m + m - 2 + 1 \leq n$. That is, $n \geq 2m - 1$. Thus the largest MCT gate can occupy $\frac{m+1}{2m-1} \times 100 \approx 50\%$ (for large m) of the n physical qubits.

As the supported gate operations on current NISQ architectures are noisy, and requires additional gates to satisfy the underlying NN-constraints, it is essential to describe the netlist using minimal number of primitive gates. We have already discussed that clean ancilla based decomposition requires less gates but more physical qubits.

Lemma 4. *In an n -qubit architecture, a r -qubit MCT netlist containing r -qubit MCT gates can be mapped if the MCT network never occupies more than 50% (or $\lfloor \frac{n+3}{2} \rfloor$) of n qubits, where clean ancilla are used in the decomposition.*

Proof. The decomposition of a r -qubit MCT gate requires $r - 3$ clean ancilla with the constraint, $r + r - 3 \leq n$. That is, $2r - 3 \leq n$. Thus the MCT netlist containing largest MCT gate can occupy, $\frac{r}{2r-3} \times 100 \approx 50\%$ (for large r) of the n physical qubits.

It may be noted that while mapping a MCT netlist to a target architecture, dirty ancilla based decomposition imposes restriction on the largest MCT gate, whereas clean ancilla based approach restricts the number of qubits. The following example illustrates the mapping of the largest MCT netlist described at primitive gate level using both clean and dirty ancilla.

Example 4. Consider the 5-qubit MCT netlist shown in Fig. 6. On the 5-qubit IBM QX2 architecture shown in Fig. 7a, the netlist can only be mapped if it is decomposed using dirty ancilla qubits (vide Lemma 3). However, on the 7-qubit IBM Falcon architecture (see Fig. 7b), such 5-qubit MCT netlists described using clean ancilla can be mapped (vide Lemma 4).



Fig. 7: IBM quantum architectures with 5 and 7 qubits.

For an n -qubit architecture, the mapping of r -qubit MCT netlist can be done by describing the network using clean ancilla if it satisfies Lemma 4; otherwise dirty ancilla should be used, provided it does not violate Lemma 3. In such cases when Lemma 4 is applicable, clean ancilla based description provides mapping with less overhead as illustrated by the following example.

Example 5. Consider the mapping of the MCT netlist shown in Fig. 6 on an IBM 7-qubit Falcon processor (see Fig. 7b). For the dirty ancilla based decomposed network, a mapping $\pi : (q_0, q_1, q_2, q_3, q_4) \rightarrow (Q_0, Q_2, Q_5, Q_1, Q_3)$ incurs a WCOST of 60. In contrast, for the clean ancilla based decomposed network, a mapping $\pi : (q_0, q_1, q_2, q_3, q_4, q_a) \rightarrow (Q_0, Q_2, Q_5, Q_6, Q_3, Q_1)$ leads to a WCOST of 48.

The choice of physical qubits as clean ancilla also plays a major role during NN-mapping to reduce gate overhead. A viable approach for effective selection of physical qubits is presented next.

3.3 Clean Ancilla Based MCT Netlist Mapping

The clean ancilla based low-level transformation of MCT netlists almost doubles the number of required physical qubits. In fact, for an n -qubit architecture, the largest MCT gate of size $n_r (= \lfloor (n+3)/2 \rfloor)$ qubits can be realized in this way by selecting required ancilla from $n_a (= n_r - 3)$ qubits. There are $\binom{n}{n_a}$ ways these n_a ancilla can be distributed among the n physical qubits.

For each ancilla configuration there exists $(n - n_a) \times \binom{n - n_a - 1}{m}$ number of $(m + 1)$ -qubit MCT gate configurations, where each MCT configuration in turn can have $\binom{n_a}{m-2}$ ancilla configurations.

For a given MCT configuration $T(C; t; A)$ where A denotes the set of ancilla qubits, the optimal remote MCT template is determined by considering $T(\pi_{min}(C); \pi_{min}(t); \pi_{min}(A))$, where π_{min} denotes mapping to physical architectural qubits, $\pi_{min} : q_i \rightarrow Q_k$ that provides minimum WCOST. By exploring all such configurations for $(m + 1)$ -qubit MCT gates where $3 \leq m + 1 \leq n_r$, a complete database of optimal remote MCT realizations is created for mapping purpose.

The MCT circuit is then processed gate-wise and each MCT gate is realized by an optimal remote MCT gate. As it has been observed in the case of dirty ancilla based decomposition [12], it is often beneficial not to directly realize an MCT gate via the corresponding remote MCT realization, but rather to apply SWAP gates first. The SWAP gates modify the mapping of logical to physical qubits, which also changes the MCT configuration of the next gate to be realized. Even though the SWAP gates seem to increase the mapping overhead, the WCOST of the resulting configuration is often significantly smaller and outweighs the cost of the SWAP gates. The same methodology as in [12] (i.e., formulation as a single source shortest path problem) is used to determine optimal combinations of SWAPs and remote MCT realization. However, in order to reduce the search space, the position of the clean ancilla qubits is fixed and those qubits will not change their position at all.

Example 6. Fig. 8 shows a clean ancilla based MCT gate, $T(\{c_1, c_2, c_3\}; t; a_1)$, realized using 3 Toffoli gates. and its corresponding QIG for the final netlist. Considering the set of clean ancilla $A = \{Q_1, Q_3, Q_6, Q_8, Q_{11}, Q_{13}, Q_{16}, Q_{18}\}$

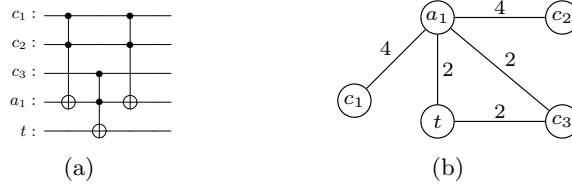


Fig. 8: (a) Clean ancilla based decomposition of a 4-qubit MCT gate using 3 Toffoli gates, and (b) corresponding interaction graph.

on IBM Q20 architecture (see Fig. 2a), the π_{min} of the remote MCT realization, $T(\{Q_2, Q_{12}, Q_{15}\}; Q_0; Q_{11})$ has WCOST of 44. Swapping the target Q_0 to Q_7 (through the shortest coupling path $Q_0 \rightarrow Q_1 \rightarrow Q_7$) as well as swapping the control Q_{15} to Q_{10} (having direct coupling $Q_{15} \rightarrow Q_{10}$) require 3 SWAPs (i.e., 9 CNOTs), but the π_{min} WCOST of the resulting configuration $T(\{Q_2, Q_{10}, Q_{12}\}; Q_7; Q_6)$ is only 20, such that the overall realization of the gate requires only $20 + 9 = 29$ CNOTs.

4 Experimental Evaluation

The proposed gate decomposition approach using clean ancilla qubits has been implemented in C++, and run on a number of reversible benchmark circuits available in RevLib [16]. For evaluation, two distinct 20-qubit architectures are considered, viz. IBM Q20 and Hexagonal Q20. We need to block 8 physical qubits for clean ancilla based decomposition; the remaining qubits can support 12-qubit MCT networks with largest MCT gate of size 11-qubits. There are $\binom{20}{8}$ possible ways in which the ancilla can be selected. For experimentation, we consider the following three clean ancilla distributions:

$$\text{Dist.1} = \{Q_1, Q_3, Q_6, Q_8, Q_{11}, Q_{13}, Q_{16}, Q_{18}\}$$

$$\text{Dist.2} = \{Q_2, Q_6, Q_7, Q_8, Q_{12}, Q_{16}, Q_{17}, Q_{18}\}$$

$$\text{Dist.3} = \{Q_1, Q_2, Q_6, Q_7, Q_{11}, Q_{12}, Q_{16}, Q_{17}\}$$

For each distribution, we generate the complete database for all n -qubit MCT gate configurations, where $3 \leq n \leq 11$. We also generate a similar dirty ancilla database for the same MCT gate configurations. For each clean ancilla distribution, database creation took under one hour whereas for dirty ancilla it took 4 hours for both the architectures. The mapping algorithm [12] took few seconds for both the architectures.

4.1 Dirty vs Clean Ancilla based Decomposition

For comparing the mapping overheads for clean and dirty ancilla based approaches, we consider the clean ancilla distribution *Dist.1*. The results of the

experiments are presented in Table 1. The first three columns give the serial no. (Sl.), the benchmark names and the number of qubits in the original netlist respectively. The next three columns respectively show the number of CNOT gates required for dirty and clean ancilla based decomposition, and the percentage improvement in using clean ancilla. Columns 6-8 show the additional CNOT gate overheads for mapping these dirty and clean ancilla based decomposed netlists on IBM Q20 architecture, and the percentage improvements respectively. Columns 9-11 show similar data for the Hexagonal Q20 architecture. The last two columns show the overall improvement percentage, calculated based on the CNOT gate overhead that includes both decomposition and mapping for clean ancilla based approach over dirty ancilla based approach for the two architectures.

The results show that clean ancilla based approach outperforms dirty ancilla based approach for most of the benchmarks. Although the mapping overhead is better for dirty ancilla based approach for many of the benchmarks, the overall improvement is better for clean ancilla based approach. Since any circuit qubits can be reused as the basis for dirty ancilla based decomposition, it is possible to have better qubit selection leading to reduced mapping overhead. On the other hand for clean ancilla based approach, distribution of ancilla greatly influences the mapping overheads across architectures.

4.2 Effect of Clean Ancilla Distribution

The overall improvements for all the three clean ancilla distributions over dirty ancilla for both the architectures are shown in Fig. 9. For both the architectures, *Dist.1* outperforms *Dist.2* and *Dist.3* in most of the cases. It is found that no single ancilla distribution across the architectures provide better results for all the benchmarks. For example, the benchmark *9symml_195* gives best result with *Dist.1*. On the other hand, for the benchmark *z4_268*, *Dist.2* provides the best result. But, *Dist.3* does not give the best result for any of the benchmarks. This clearly shows that the distribution of clean ancilla has a significant impact on the mapping overhead.

5 Conclusion

In this work we critically analyze two alternate decomposition schemes for realizing MCT netlists using Clifford+T gates based on (i) dirty ancilla, and (ii) clean ancilla. With the increasing power of modern-day quantum computers we can use clean ancilla based decomposition that reduces the overall gates in the transpiled netlist, which in turn increases operation fidelity. Two alternate 20-qubit architectures have been used for transpilation, the IBM Q20 and the 20-qubit hexagonal architectures. We show that although dirty ancilla help in reducing the total number of required qubits, the use of clean ancilla can significantly reduce the number of gate operations and hence the accumulated errors. In addition, using a database-driven approach the impact of distributing clean ancilla for NN-mapping of logical qubits has been analyzed across architectures.

Table 1: Results of clean and dirty ancilla based decomposition for IBM and Hexagonal architectures

Sl.	Benchmark	n	MCT decomposition				IBM Q20 Mapping Ovrd.				Hex. Q20 Mapping Ovrd.				Overall Imp.(%)	
			dirty	clean	Imp.(%)		dirty	clean	Imp.(%)		dirty	clean	Imp.(%)		IBM	HEX
1	9symml_195	10	6792	3152	53.59	1380	954	30.87	1026	1002	2.34	49.76	46.87			
2	alu-v2_30	5	146	114	21.92	30	84	-180.00	27	90	-233.33	-12.50	-17.92			
3	cm152a_212	12	358	218	39.11	27	66	-144.44	27	87	-222.22	26.23	20.78			
4	con1_216	9	284	180	36.62	45	98	-117.78	45	118	-162.22	15.50	9.42			
5	cycle10_2_110	12	1264	616	51.27	261	283	-8.43	267	335	-25.47	41.05	37.88			
6	dc1_220	11	570	390	31.58	105	354	-237.14	105	373	-255.24	-10.22	-13.04			
7	dc1_221	11	570	390	31.58	111	368	-231.53	99	368	-271.72	-11.31	-13.30			
8	f2_232	8	358	222	37.99	39	99	-153.85	36	123	-241.67	19.14	12.44			
9	rd73_252	10	1468	876	40.33	306	922	-201.31	273	1008	-269.23	-1.35	-8.21			
10	sym10_262	11	12034	5514	54.18	2742	2103	23.30	2358	2205	6.49	48.45	46.37			
11	sym6_145	7	1008	600	40.48	129	252	-95.35	117	285	-143.59	25.07	21.33			
12	sym9_148	10	6300	3948	37.33	576	2750	-377.43	489	3080	-529.86	2.59	-3.52			
13	sym9_193	10	6792	3152	53.59	1398	957	31.55	1041	1023	1.73	49.83	46.70			
14	urf1_150	9	59133	33461	43.41	11323	22044	-94.68	10230	23588	-130.58	21.22	17.75			
15	urf1_151	9	56583	32187	43.12	10929	21911	-100.48	10040	22861	-127.70	19.87	17.37			
16	urf2_153	8	19951	11899	40.36	3784	5747	-51.88	3471	6539	-88.39	25.65	21.28			
17	urf2_154	8	18857	11289	40.13	3733	5717	-53.15	3485	6334	-81.75	24.72	21.12			
18	wim_266	11	290	202	30.34	48	172	-258.33	51	181	-254.90	-10.65	-12.32			
19	z4_268	11	920	560	39.13	108	402	-272.22	150	437	-191.33	6.42	6.82			

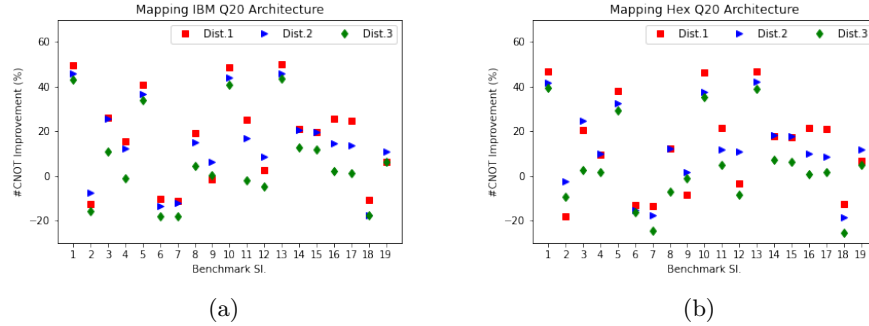


Fig. 9: Effects of ancilla distribution across architectures.

References

1. IBM Q. <https://www.research.ibm.com/ibm-q>, [Accessed: 2019-03-20]
2. IBM Quantum Processor Types. <https://quantum-computing.ibm.com/services/docs/services/manage/systems/processors/> (2017)
3. The IBM Hummingbird Architecture. <https://quantum-computing.ibm.com/services/docs/services/manage/systems/processorshummingbird/> (2021), [Online; accessed 1-December-2021]
4. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* **32**(6), 818–830 (June 2013)
5. Baker, J.M., Duckering, C., Hoover, A., Chong, F.T.: Decomposing quantum generalized toffoli with an arbitrary number of ancilla. *ArXiv* **abs/1904.01671** (2019)
6. Balauca, S., Arusoae, A.: Efficient constructions for simulating multi controlled quantum gates. In: *Computational Science – ICCS 2022*. pp. 1–14 (2022)
7. Barenco, A., Bennet, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. *Physical Review A* **52**(5), 3457–3467 (Nov 1995)
8. Chow, J., Dial, O., Gambetta, J.: IBM Quantum breaks the 100-qubit processor barrier. <https://research.ibm.com/blog/127-qubit-quantum-processor-eagle/> (2021), [Online; accessed 16-November-2021]
9. Kole, A., Hillmich, S., Datta, K., Wille, R., Sengupta, I.: Improved mapping of quantum circuits to IBM QX architectures **39**(10), 2375–2383 (2020)
10. Nation, P., Paik, H., Cross, A., Nazario, Z.: The IBM Quantum heavy hex lattice. <https://research.ibm.com/blog/heavy-hex-lattice/> (2021), [Online; accessed 7-July-2021]
11. Nielsen, M., Chuang, I.: *Quantum Computation and Quantum Information*. Cambridge Univ. Press (Oct 2000)
12. Niemann, P., Bandyopadhyay, C., Drechsler, R.: Improved NCV gate realization of arbitrary size Toffoli gates. In: *Design, Automation and Test in Europe (DATE)*. pp. 200–205 (Feb 2021)
13. Rahman, M., Dueck, G.W.: Synthesis of linear nearest neighbor quantum circuits. In: *10th Int’l Workshop on Boolean Problems*. pp. 1–9 (2012)
14. Selinger, P.: Quantum circuits of t-depth one. *Physical Review A* **87**(4) (apr 2013)

15. Tang, H., et al.: Experimental quantum fast hitting on hexagonal graphs. *Nature Photonics* **12**(12), 754–758 (2018)
16. Wille, R., Große, D., Teuber, L., Dueck, G., Drechsler, R.: RevLib: An on-line resource for reversible functions and reversible circuits. In: *Proc. Intl. Symp. on Multi-Valued Logic*. pp. 220–225 (2008), RevLib is available at <http://www.revlib.org>
17. Wille, R., Lye, A., Drechsler, R.: Optimal SWAP gate insertion for nearest neighbor quantum circuits. In: *Proc. Design Automation Conference (DAC)*. pp. 489–494. IEEE, Suntec, Singapore (2014)
18. Zulehner, A., Paler, A., Wille, R.: An efficient methodology for mapping quantum circuits to the IBM QX architectures **38**(7), 1226–1236 (2019)