Unlocking the Benefits of Dynamic Quantum Circuits in Resource Constraint Architecture

Abhoy Kole[△], Mohammed E. Djeridane[S,𝕌], Arighna Deb[K], Kamalika Datta[△,𝕌],
Indranil Sengupta[I], Rolf Drechsler[△,𝕌]
[△]*German Research Centre for Artificial Intelligence (DFKI), Bremen, Germany*
[S]*Siemens EDA GmbH, Hamburg, Germany*
[K]*School of Electronics Engineering, KIIT DU, India*
[I]*Department of Computer Science and Engineering, IIT Kharagpur, India*
[𝕌]*Institute of Computer Science, University of Bremen, Germany*
abhoy.kole@dfki.de, djeridan@uni-bremen.de, airghna.debfet@kiit.ac.in, kdatta@uni-bremen.de,
isg@iitkgp.ac.in, drechsler@uni-bremen.de

***Abstract*—The execution of large-scale quantum algorithms is currently constrained by the limited number of available qubits, qubit connectivity restrictions, and the inherent noise in quantum processors. To address these limitations, a design methodology known as Dynamic Quantum Circuits (DQC) has emerged. DQC leverages non-unitary operations—such as active reset, mid-circuit measurement, and classically controlled gate operations—to reduce qubit requirements during circuit design. Recently, DQC-based implementations have been explored for various algorithms, including Shor's Prime Factorization, Quantum Phase Estimation (QPE), and Bernstein-Vazirani (BV), as well as for key operations like state preparation, Toffoli networks, and non-local gates. While DQC offers a significant reduction in qubit usage, it introduces a trade-off in the form of increased circuit depth. Therefore, assessing the reliability of such circuits becomes crucial in the context of current quantum hardware architectures. In this paper, we analyze the reliability of DQC-based quantum circuit realizations as a function of qubit count and circuit depth. We present empirical results for two algorithms and evaluate how architectural parameters impact their reliability.**

***Index Terms*—Dynamic Quantum Circuits, Quantum Algorithms, Reliability and Fidelity**

1. Introduction

Recent advancements in quantum computing have highlighted its potential to tackle computationally hard problems, leading to the development of small to medium-scale quantum processors and a growing focus on demonstrating quantum supremacy [1]. However, executing quantum algorithms remains challenging due to the limited number of physical qubits and coupling constraints within target quantum processors. The principle of deferred measurement enables the incorporation of non-unitary operations—such as active reset, mid-circuit measurement, and classically controlled gate operations—into quantum circuits, giving rise to Dynamic Quantum Circuits (DQC). This approach allows for design trade-offs, such as reducing qubit requirements by recycling them for subsequent operations (e.g., [2]).

Numerous studies have investigated DQC-based design methodologies and successfully implemented them on real quantum hardware, demonstrating their potential for constructing quantum circuits in resource-constrained environments. These efforts span a range of applications, including quantum algorithms [3], [4], Toffoli gate operations [5], [6], quantum state preparation [7], and non-local gate implementations [8]. Notably, for algorithms such as Shor's prime factorization [2], Quantum Phase Estimation (QPE) [3], and the Bernstein-Vazirani (BV) algorithm [9], DQC techniques enable implementations using only a single input qubit, regardless of the number of input or result qubits originally required.

However, this qubit-efficient realization comes at the cost of increased circuit depth, which in turn leads to higher latency. Since quantum circuits are susceptible to errors from both gate operations and decoherence, deeper circuits tend to suffer from reduced computational reliability. To address this challenge, a study presented in [10] proposes an alternative realization of Shor's algorithm that minimizes the number of input qubits while also optimizing circuit depth, making it suitable for both monolithic and distributed quantum computing environments.

In this work, we present a study on a dynamic transformation scheme aimed at minimizing the depth of quantum circuits. Our approach highlights the importance of incorporating hardware-specific information, enabling an architecture-aware DQC-based transformation followed by gate mapping. This facilitates a performance trade-off between circuit depth and qubit count, allowing the algorithm to be tuned based on the number of available qubits—more qubits result in shallower circuits. We further analyze how variations in circuit depth impact the fidelity of quantum circuits. Experimental results are provided for several well-known quantum algorithms, including BV [11], QPE [12], and Deutsch-Jozsa (DJ) [13].

The rest of this paper is organized as follows: Section II provides a brief overview of quantum circuits, NISQ-era processors, and the principles of DQC. Section III details the proposed DQC transformation under resource constraints. Section IV presents the experimental results, followed by concluding remarks in Section V.

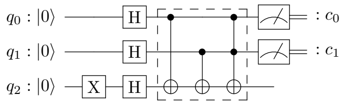


Figure 1: A 3-qubit quantum circuit realizing DJ algorithm for evaluating a black-box function $\mathcal{F}(a, b) = a + b$.

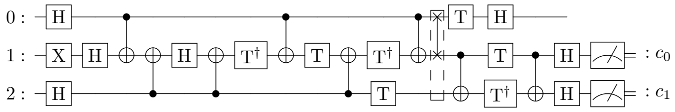


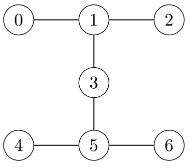
Figure 3: DJ algorithm described using *Clifford+T* gates for evaluating the function $\mathcal{F}(a, b) = a + b$ using IBM Lagos processor.

2. Background

2.1. Quantum Circuit Model

In quantum computing, we carry out a sequence of gate operations on a set of qubits that are often classified as data, answer or ancilla qubits. Initially all the qubits are set to some computational basis state ($|0\rangle$ or $|1\rangle$), and during computation their states evolve and may goes in superposition, e.g. $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, or entangled with the state of some other qubit, e.g. $|\phi\rangle = \alpha |00\rangle + \beta |11\rangle$, where α and β are complex amplitudes such that $|\alpha|^2 + |\beta|^2 = 1$. Finally to obtain the result, the qubit states are measured that causes the states like $|\psi\rangle$ and $|\phi\rangle$ to settle down into one of the basis states $|0\rangle$ or $|1\rangle$, and $|00\rangle$ or $|11\rangle$, with probabilities $|\alpha|^2$ or $|\beta|^2$, respectively. Such computation can have the edges over classical computing in certain applications.

Typically, quantum algorithms are described as circuits, using gates of various types such as 1-qubit $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$ gates, Pauli gates (X, Y and Z), Phase gates ($P(\theta)$, S and T) and Hadamard gate (H), and multi-qubit gates like Toffoli, *CNOT*, Fredkin and *SWAP*. The inverse of any such gate U is often denoted by $U^\dagger$, i.e. $UU^\dagger = U^\dagger U = I$. The *Clifford+T* gate library [14] that comprises of H, X, $T/T^\dagger$ and *CNOT* gates, is the most common choice due to its fault-tolerance capability [15]. Fig. 1 shows a quantum circuit with three qubits realizing the *Deitsch-Jozsa* (DJ) algorithm to evaluate the constant/balance nature of a black-box function $\mathcal{F}(a, b) = a + b$ surrounded by the dashed-rectangle.



(a)

CNOT	9.014e−3 (ϵ)
U	4.596e−4 (ϵ)
⟋	1.504e−2 (ϵ)
T_1	145.37 μs
T_2	72.21 μs

(b)

Figure 2: (a) The coupling-graph of a 7-qubit IBM Lagos quantum processor, and (b) average gate and measurement error rates as well as qubit decoherence time.

2.2. NISQ-era Processors

Today's quantum processors have small number of noisy qubits that limits the size of quantum algorithms that can be executed, are also susceptible to errors in producing the desired outcome. This is due to imperfections in realizing gate and measurement operations as well as retaining the state of the qubits for a reasonable time period. Fig. 2a shows the coupling-graph of a 7-qubit IBM Falcom processor, IBM Lagos. The gate set supported by the device is $\{R_Z(\theta), X, \sqrt{X}, \mathit{CNOT}\}$. The gate and measurement error rates as well as coherence period differ from qubit to qubit as shown in Fig. 2b.

In order to run any quantum algorithm on a Noisy Intermediate-Scale Quantum (NISQ) device, the algorithm must be expressed using basis gates, at the same time satisfying the qubit coupling constraints of the processor. There are various approaches (e.g. [16]) that can be used to obtain the basis gate description from the initial *Clifford+T* realization by replacing individual gate operation with their equivalent basis gate description (e.g. H and $T/T^\dagger$ operations from *Clifford+T* library can be replaced with $R_Z(\frac{\pi}{2})\sqrt{X}R_Z(\frac{\pi}{2})$ and $R_Z(\frac{\pi}{4})/R_Z(-\frac{\pi}{4})$ respectively).

The violation of coupling constraints is often compensated using additional *SWAP* gates [17]. The *SWAP* gates further increases the depth of the circuit and also compromises the operational fidelity. Fig. 3 shows the Clifford+T description of the DJ algorithm for the black-box $\mathcal{F}(a, b) = a + b$ (see Fig. 1) for the mapping: $(q_0, q_1, q_2) \mapsto (0, 2, 1)$. A *SWAP* gate (shown as the dashed rectangle) is inserted before the final two *CNOT* operations to address the coupling-map violation of the IBM Lagos processor (see Fig. 2a) for the present mapping.

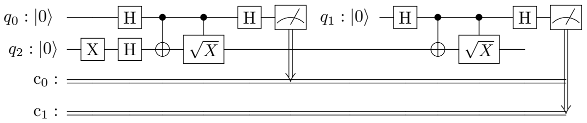


Figure 4: A 2-qubit dynamic description of DJ algorithm for evaluating the black-box function $\mathcal{F}(a, b) = a + b$ using unrolled dynamic Toffoli realization [9].

2.3. Dynamic Transformation Problem

The conventional quantum circuit structure described earlier serves as the basis for implementing quantum algorithms on physical quantum devices. This approach is static and cannot adapt computations based on interim results. Dynamic Quantum Circuit (DQC) allows certain classical computing instructions to be executed on quantum computers, enabled by non-unitary operations like mid-circuit measurements, active-reset, and classical input-controlled gate operations. Recent studies have highlighted the benefits of DQCs by demonstrating dynamic versions of Shor, QPE and BV algorithms using fewer qubits [2], [3]. Another study explores converting the Toffoli gate into its dynamic counterpart [9], and also discusses dynamic implementations of 3-qubit DJ circuits using two distinct dynamic Toffoli gate designs referred to as *dynamic 1* and *dynamic 2*.

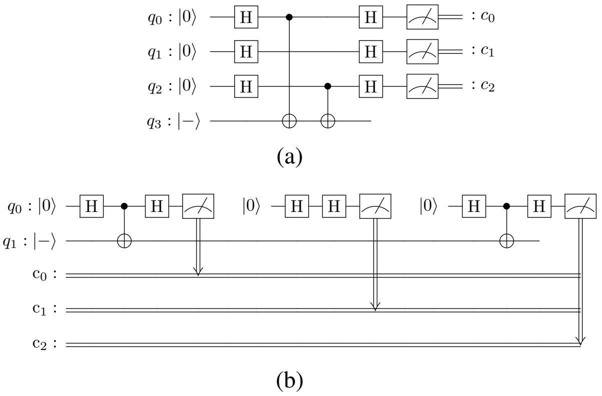


Figure 5: (a) A 4-qubit BV circuit for detecting the hidden binary string 101, and (b) its dynamic realization using two qubits.

However, dynamic realizations of quantum circuits leads to an increase in circuit depth. For example, the dynamic realization of a 3-qubit function using two qubits has a circuit depth of 11 as shown in Fig. 4, while the traditional version has a depth of 6 (see Fig. 1). The increase in depth due to the feed-forward nature of DQC presents a challenge when implementing them on real quantum devices.

3. Resource Constraint Transformation

The dynamic transformation of quantum circuits—shaped by architectural constraints such as qubit availability, coupling constraints and gate error rates—depends heavily on the specific characteristics of the circuit. In this work, we analyze two representative cases and propose transformation strategies optimized for limited resources.

3.1. Case I: Without Gate Dependency

Quantum algorithms are often designed considering the data $|X\rangle$ and answer $|Y\rangle$ qubits separately. The controlled unitary operations like $CU(X, Y)$ are then conducted on answer $|Y\rangle$ depending on the state of data $|X\rangle$, i.e. $U\,|Y\rangle$ if $|X\rangle = |11\cdots 1\rangle$; $I\,|Y\rangle$ otherwise. In presence of more than one such CU operations, the dynamic transformation can be realized using only two qubits when $|Y\rangle$ represent a single qubit state. The following example illustrates the BV algorithm with more than one $CNOT$ operations.

Example 1. *Consider the quantum circuit shown in Fig. 5a for detecting the unknown binary string* 101 *from a black-box function using BV algorithm. The qubits q_0, q_1 and q_2 represent* 3*-qubit data $|X\rangle$ initially at state $|000\rangle$ while q_3 represents the single-qubit answer $|Y\rangle = |-\rangle$. The outcome is obtained by measuring the three data qubits q_0, q_1 and q_2 at the end of execution into three classical single-bit registers c_0, c_1 and c_2, respectively.*

When there is no dependency (non-commutativity) among the control unitaries, the interaction between one of the data qubit and the answer qubit can be conducted independently in any order. This does not sacrifice the execution reliability as long as the measured outcomes are stored in appropriate classical registers as illustrated in the following example.

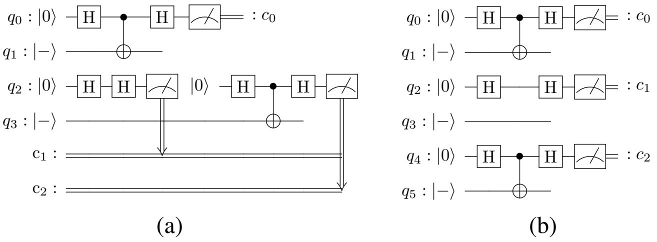


Figure 6: (a) A 4-qubit dynamic realization of the previously considered BV algorithm for detecting the hidden binary string 101, and (b) its equivalent 6-qubit dynamic realization of minimal depth.

Example 2. *Consider again the circuit shown in Fig. 5a, and its 2-qubit realization shown in Fig. 5b. While q_1 is dedicated to the single-qubit answer $|Y\rangle$ initialized to $|-\rangle$ during the entire execution of the circuit, q_0 is assigned the role of one of the three data qubits of actual circuit (see Fig. 5a) interchangeably. In between the transition of role, q_0 is measured in corresponding classical register using mid-circuit measurement operation before applying active reset operation to prepare it for the next unassigned data qubit.*

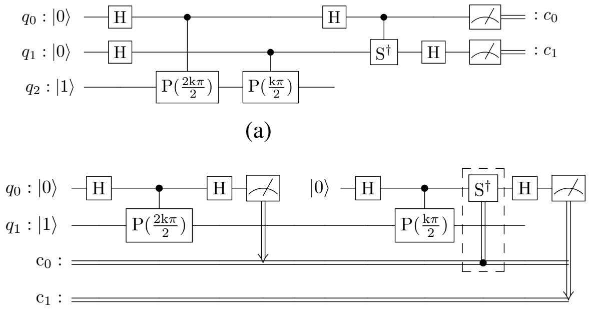
The dynamic transformation approach addresses the area requirements (in terms of number of qubits) and eliminates the chances of additional noise due to the *SWAP* overhead. However, the circuit depth corresponding to the answer qubits $|Y\rangle$ remain a concern as qubits suffer from limited coherence period. The depth of answer qubit interaction can be minimized by replicating it more than once depending on its availability in the target architecture, without affecting the execution outcome in certain cases when the answer $|Y\rangle$ represent the *eigenstate* like the BV algorithm. The depth minimization of answer $|Y\rangle$ is discussed in the following example.

Example 3. *Consider the mapping of the circuit of Fig. 5a on a superconducting QPU with two disjoint set of coupled qubit pairs (q_0, q_1) and (q_2, q_3). This allows replicating answer $|Y\rangle$ twice on q_1 and q_3, while allocating one of the three data qubits representing $|X\rangle$ to q_0 and other two to q_2 as shown in Fig. 6a. This leads to reduction of maximum interaction depth of answer qubit from* 11 *(see Fig. 5b) to* 6 *(see Fig. 6a). For simplicity of depth estimation, we ignore the gate overhead to prepare answer $|Y\rangle$ to $|-\rangle$ and consider all the gate operations performed on both data and answer qubits till their final interaction. Fig. 6b shows the maximum reduction of interaction depth of answer qubit to 4, by replicating using three disjoint set of coupled qubit pairs (q_0, q_1), (q_2, q_3) and (q_4, q_5).*

The major benefits of such dynamic configuration is three-fold, (1) it reduces the overall depth of the transformed quantum circuit, (2) it eliminates the requirement of *SWAP* operations for executing on a coupling restricted architecture, and (3) can be realized without DQC supprts.

3.2. Case II: In Presence of Gate Dependency

Quantum algorithms may have gate dependencies due to interactions among data qubits, and require strict ordering



(a)

(b)

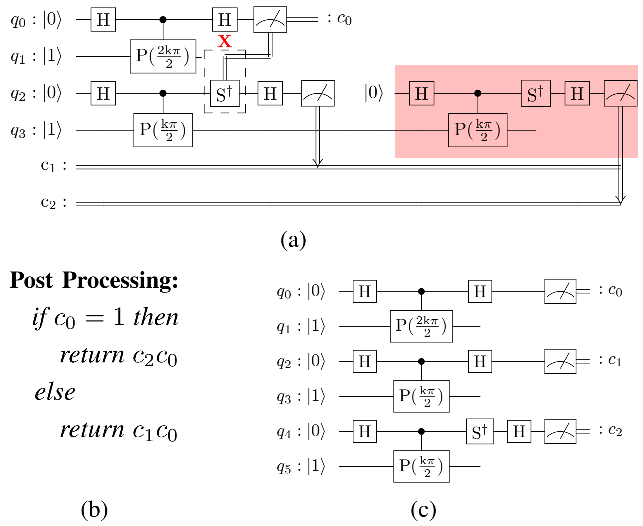
Figure 7: (a) 3-qubit QPE algorithm for estimating the phase of gate operation $P(\frac{k\pi}{2})$ for any integer $k \geqslant 0$ in a 2-bit classical register c_1c_0, and (b) its 2-qubit dynamic realization.

of data qubits for dynamic transformation. The following example illustrates the dynamic transformation of QPE for estimating phase of $P(\frac{k\pi}{2})$ for any integer $k \geqslant 0$ using two qubits.

Example 4. *Fig. 7a shows a* 3*-qubit circuit for estimating the phase of gate operation* $P(\frac{k\pi}{2})$ *for any integer* $k \geqslant 0$*. The qubits* q_0 *and* q_1 *represent data* $|X\rangle$*, while* q_2 *indicates the single-qubit answer* $|Y\rangle$*. At the end of the execution,* q_0 *and* q_1 *are measured in two single-bit classical registers* c_0 *and* c_1*, respectively to obtain the phase value as binary string* c_1c_0*. The corresponding dynamic realization using two qubits is shown in Fig. 7b. The interaction among data qubits in the form of* $CS^\dagger$ *causes a strict order of allocating* q_0 *of the* 3*-qubit realization (see Fig. 7a) before allocation of* q_1 *where the interaction is realized as the classically controlled* $S^\dagger$ *operation.*

The transformation results in an increase in circuit depth, while achieving area (qubit) minimization and avoiding *SWAP* operations. With more number of disjoint coupled qubit pairs available, the dynamic transformation can result in further reduction in interaction depth. This depth optimal realization cannot be achieved in a straightforward fashion, like the one discussed earlier for BV algorithm. Furthermore, in the absence of support for DQC operations, the realization can still be obtained using (i) additional classical registers to store the measured outcome, followed by (ii) post processing of the classical data to obtain the desired outcome. The approach is illustrated in the following example.

Example 5. *Consider the dynamic transformation of QPE algorithm as shown in Fig. 7b. Using two disjoint pairs of coupled qubits (*q_0*,* q_1*) and (*q_2*,* q_3*), the data* $|X\rangle$ *and answer* $|Y\rangle$ *qubit interaction can be caried out independently as shown in Fig. 8a. In the absence of classically controlled gate operations (marked as X), the realization replaces the classically controlled* $S^\dagger$ *operation (surrounded by dashed rectangle) with 1-qubit unitary* $S^\dagger$ *operation by repeating the sub-circuit instance (highlighted in red color) one more time. This requires* 3*-bit classical register instead of* 2*-bit. The dynamic transformation requires a post processing, i.e. verifying the measured outcome of classical register* c_0*, the second bit is decided as either* c_2 *(when* $c_0 = 1$*) or* c_1 *(when* $c_0 = 0$*) as shown in Fig. 8b. The depth of the netlist can*



(a)

```
Post Processing:
  if $c_0 = 1$ then
      return $c_2c_0$
  else
      return $c_1c_0$
```

(b)

(c)

Figure 8: (a) A 4-qubit dynamic transformation of QPE algorithm for 2-bit phase estimation of gate operation $P(\frac{k\pi}{2})$ for any integer $k \geqslant 0$, (b) the required post processing of classical outcome, and (c) 6-qubit depth-optimal dynamic realization.

be reduced further by allocating one additional coupled qubit pair to one of the split iteration due to the interaction between the second data qubit and answer qubit as depicted in Fig. 8c. This reduces the depth of the answer qubit from 7 *to* 2 *ignoring the initial state preparation for the answer qubit* $|Y\rangle$*.*

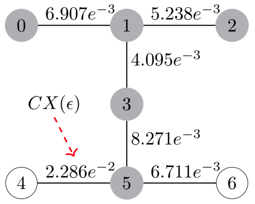
The approach functions effectively in the absence of classically controlled unitary operations; however, it may lead to exponential sub-circuit executions due to the repeated allocation of dependent data qubits—specifically, those serving as targets of controlled unitary operations—twice for each such dependency. Furthermore, the design complexity can be mitigated through the use of classically controlled unitaries, but the resulted circuit also enhance the reliability of QPE simulations on hardware lacking support for DQCs.

4. Experimental Evaluation

For evaluation of the proposed DQC transformation, we have conducted two set of experiments: (i) analyze the resource requirements in terms of qubit count, circuit depth and gate count, and (ii) assess the operational fidelity for execution on a coupling restricted noisy quantum processor without support for classical control gate operations. We have considered the following quantum circuits as benchmarks realizing BV [11], QPE [12] and DJ [13] algorithms:

1) $BV\_a_2a_1a_0$ denotes the benchmark realizing BV algorithm for the 3-bit secret binary string $a_2a_1a_0$.
2) $QPE\_p_1p_0$ indicates the benchmarks realizing QPE algorithm to determine the 2-bit phase of the operator $P(\frac{k\pi}{2})$ for $k = 1, 2, 3, 4$.
3) $DJ\_\mathcal{F}$ represents the benchmarks realizing DJ algorithm to determine the constant-balance property of 2-bit functions of the form $\mathcal{F}(a, b)$ that require at least one Toffoli operation for implementation.

The two-qubit dynamic realization of these benchmarks are



Q	$T_1(\mu s)$	$T_2(\mu s)$	$U(\epsilon)$	⏲ (ϵ)
0	107.66	41.45	$1.993e^{-4}$	$1.080e^{-2}$
1	137.28	84.51	$1.679e^{-4}$	$1.590e^{-2}$
2	142.92	69.68	$2.047e^{-4}$	$7.600e^{-3}$
3	123.96	68.17	$1.462e^{-4}$	$1.430e^{-2}$
4	147.52	27.54	$2.152e^{-3}$	$2.570e^{-2}$
5	171.40	78.02	$1.743e^{-4}$	$1.480e^{-2}$
6	186.87	118.09	$1.727e^{-4}$	$1.620e^{-2}$

Figure 9: The qubits selected for the experiment and the noise model of the IBM Lagos quantum processor.

derived from the prior works [3], [9]. Due to the operational accuracy, the unrolled version of dynamic Toffoli operation proposed in [9] is considered for dynamic realization of the DJ algorithm. In order to make a uniform comparison, the same basis gate set $\{H, X, P(\theta), CNOT\}$ is used to describe all the benchmarks. Finally, for verifying the correctness and evaluating the corresponding fidelity of these benchmarks, the *Qiskit SDK* [18] version 0.43.3 and a Falcon family 7-qubit IBM Lagos quantum processor are used in the experiments.

4.1. Assessment of Resources

Table 1 compares traditional and dynamic implementations (optimized for minimal qubits and minimal depth) of the BV, QPE, and DJ benchmarks. The first column lists benchmark names, followed by gate counts—single-qubit (U), two-qubit (CX)—-and overall depth for traditional implementations. These use 4 qubits for BV and 3 for QPE and DJ.

The next five columns show DQC transformation results for minimal qubits (based on [3], [9]), along with improvements in gate count (Δ_G) and depth (Δ_D). The final five columns present depth-optimized dynamic results. It can be verified from the table that:

a) Minimal qubit designs (using only 2 qubits) can increase depth significantly (up to $\sim 160\%$).
b) Depth-optimized designs require more qubits (e.g., 6 for 4-qubit BV) but reduce depth by up to $\sim 53\%$.

These results illustrate a trade-off between qubit count and circuit depth, relevant for execution on noisy quantum hardware with limited resources. The next section evaluates this trade-off in more detail.

4.2. Evaluation of Design Effectiveness

To evaluate the effectiveness of the proposed dynamic design extension, we utilize the Qiskit Aer Simulator [18] along with the noise model of the IBM Lagos device, as illustrated in Fig. 9. Based on minimal gate and measurement error rates, the following qubit configurations were selected for mapping the three benchmark classes:

$$\begin{array}{rll} \text{BV:} & (q_0, q_1, q_2, q_3) \mapsto (0, 2, 3, 1) & \multirow{2}{*}{$\Big\}\text{Trad.}$} \\ \text{QPE, DJ:} & (q_0, q_1, q_2) \mapsto (2, 3, 1) & \\ \text{FOR ALL:} & (q_0, q_1) \mapsto (2, 1) & \text{Dyn. Area} \\ \text{FOR ALL:} & (q_0, q_1, q_2, q_3) \mapsto (2, 1, 3, 5) & \text{Dyn. Depth} \end{array}$$

Due to the lack of three disjoint pairs of coupled qubits in IBM Lagos (see Fig. 9), the 4-qubit depth-minimized dynamic version was used for this experiment. Notably, under the chosen mapping ($q_i \mapsto k$), only the traditional benchmarks require additional SWAP gates, leading to increased circuit depth, as illustrated in Fig. 10.

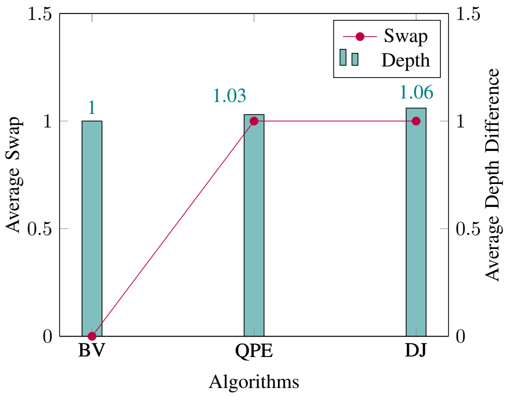


Figure 10: Average *SWAP* requirements and depth increase in mapping traditional BV, QPE and DJ benchmarks on IBM Lagos quantum processor.

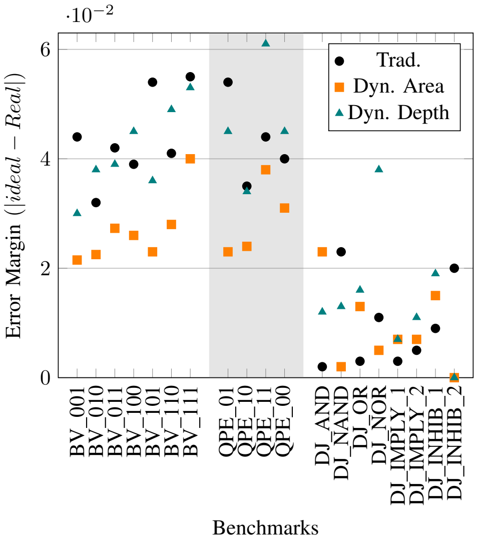


Figure 11: Fidelity analysis of running traditional, and area and depth minimal BV, QPE and DJ benchmarks on IBM Lagos quantum processor.

Fidelity analysis for all three benchmark classes is presented in Fig. 11. For the BV and QPE benchmarks, the results clearly demonstrate improved fidelity with dynamic configurations. However, this distinction is not observed in the DJ benchmarks due to the specific functions used, none of which are truly balanced or constant. Even in an ideal environment, these functions evaluate to a constant with $\approx 25\%$ probability, consistent with findings in [9].

Additionally, the dynamic area-minimized versions of BV and QPE benchmarks show superior fidelity compared to both their depth-minimized dynamic and traditional counterparts. This outcome is attributed to the qubit selection: the area-minimized versions use qubits (2, 1), which have lower gate and measurement error rates than those used in the depth-minimized (2, 1, 3, 5) and traditional (0, 2, 3, 1)/(2, 3, 1) configurations (see Fig. 9).

TABLE 1: Comparison of traditional and resource-optimized (i.e, area (qubits) and depth minimal) realizations of BV, QPE, and DJ algorithms.

Benchmark	**Trad.** ($n = 4/3^\star$)			**Dyn. Area** ($n = 2$) [3], [9]					**Dyn. Depth** ($n = 6/4^*$)				
	U	CX	D	U	CX	D	$\Delta_G(\%)$	$\Delta_D(\%)$	U	CX	D	$\Delta_G(\%)$	$\Delta_D(\%)$
BV_001	8	1	5	8	1	13	0	-160	12	1	5	-44.44	0
BV_010	8	1	5	8	1	12	0	-140	12	1	5	-44.44	0
BV_011	8	2	6	8	2	14	0	-133.33	12	2	5	-40	16.67
BV_100	8	1	5	8	1	12	0	-140	12	1	5	-44.44	0
BV_101	8	2	6	8	2	14	0	-133.33	12	2	5	-40	16.67
BV_110	8	2	6	8	2	13	0	-116.67	12	2	5	-40	16.67
BV_111	8	3	7	8	3	15	0	-114.29	12	3	5	-36.36	28.57
QPE_01	13	5	14	11	3	14	22.22	0	18	5	8	-27.78	42.86
QPE_10	13	5	15	11	3	13	22.22	13.33	17	4	7	-16.67	53.33
QPE_11	14	6	16	12	4	16	20	0	19	6	8	-25	50
QPE_00	14	6	16	12	4	16	20	0	19	6	8	-25	50
DJ_AND	15	6	15	16	4	17	4.76	-13.33	18	4	9	-4.76	40
DJ_NAND	16	6	16	17	4	18	4.54	-12.5	19	4	10	-4.55	37.5
DJ_OR	15	8	17	16	6	20	4.35	-17.65	18	6	10	-4.35	41.18
DJ_NOR	16	8	18	17	6	21	4.17	-16.67	19	6	11	-4.17	38.89
DJ_IMPLY_1	16	7	17	17	5	20	4.35	-17.64	19	5	10	-4.35	41.18
DJ_IMPLY_2	16	7	17	17	5	19	4.35	-11.76	19	5	11	-4.35	35.29
DJ_INHIB_1	15	7	16	16	5	18	4.54	-12.5	18	5	10	-4.54	37.5
DJ_INHIB_2	15	7	16	16	5	19	4.54	-18.75	18	5	10	-4.54	37.5

$n \rightarrow$ Number of qubits; $4/3^\star \rightarrow$ For BV $n = 4$ while for QPE and DJ $n = 3$; $U \rightarrow$ Number of 1-qubit gates; $CX \rightarrow$ Number of 2-qubit gates; $D \rightarrow$ Depth; Δ_G / $\Delta_D \rightarrow$ Improvement in gate count / depth of dynamic over traditional; $6/4^* \rightarrow$ For BV and QPE $n = 6$ while for DJ $n = 4$.

5. Conclusion

This work investigates DQC-based transformation schemes, analyzing their dependence on qubit count and circuit depth, and their effect on reliability in NISQ processors. Traditional circuits face hardware constraints such as limited qubit counts and restrictive coupling maps. DQC-based designs offer a flexible alternative often at the cost of increased depth and reduced fidelity. To mitigate this issue, we explored depth-optimized DQC-based designs and demonstrated their effectiveness using benchmark circuits such as BV, QPE, and DJ. Our results show that this approach enhances reliability by reducing circuit depth, even on devices that do not natively support DQC—though at the expense of increased design complexity. Future work will aim to extend this methodology to support higher-order quantum gates and more complex algorithms, paving the way for broader applicability in practical quantum computing scenarios.

Acknowledgment

This research has been partly supported by the Federal Ministry of Research, Technology and Space (BMFTR, formerly BMBF) within the project EASEPROFIT under grant no. 16KIS2127.

References

[1] F. A. et al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, oct 2019.

[2] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'Brien, "Experimental realization of shor's quantum factoring algorithm using qubit recycling," *Nature Photon*, vol. 6, no. 11, pp. 773–776, publisher: Nature Publishing Group.

[3] A. Córcoles *et al.*, "Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits," *Physical Review Letters*, vol. 127, no. 10, 2021.

[4] A. Kole, K. Datta, and R. Drechsler, "Exploring the potential of dynamic quantum circuit for improving device scalability," in *2024 IEEE 37th International System-on-Chip Conference (SOCC)*, 2024, pp. 1–5.

[5] A. Kole, A. Deb, K. Datta, and R. Drechsler, "Dynamic realization of multiple control toffoli gate," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, ISSN: 1558-1101.

[6] E. Bäumer, V. Tripathi, D. S. Wang *et al.*, "Efficient long-range entanglement using dynamic circuits," *PRX Quantum*, vol. 5, no. 3, p. 030339, publisher: American Physical Society.

[7] S. A. Moses, C. H. Baldwin, M. S. Allman *et al.*, "A race-track trapped-ion quantum processor," *Phys. Rev. X*, vol. 13, p. 041052, Dec 2023.

[8] Y. Wan, D. Kienzler, S. D. Erickson *et al.*, "Quantum gate teleportation between separated qubits in a trapped-ion processor," *Science*, vol. 364, no. 6443, pp. 875–878.

[9] A. Kole, A. Deb, K. Datta, and R. Drechsler, "Extending the design space of dynamic quantum circuits for toffoli based network," in *Design, Automation and Test in Europe (DATE)*, 2023, pp. 1–6.

[10] M. Schmidt, A. Kole, L. Wichette *et al.*, "Exploration of design alternatives for reducing idle time in Shor's algorithm: A study on monolithic and distributed quantum systems," no. arXiv:2503.22564.

[11] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1411–1473, 1997.

[12] A. Y. Kitaev, "Quantum measurements and the Abelian stabilizer problem," 11 1995.

[13] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," in *Proceedings of the Royal Society of London*, vol. 439, 1992, pp. 553–558.

[14] H. N. N.H. Nickerson, Y. Li, and S. Benjamin, "Topological quantum computing with a very noisy network and local error rates approaching one percent," *Nature Communications*, vol. 4, p. 1756, 2013.

[15] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," *Nature*, vol. 549, no. 7671, pp. 172–179, Sep 2017.

[16] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. on CAD*, vol. 32, no. 6, pp. 818–830, June 2013.

[17] A. Kole, S. Hillmich, K. Datta, R. Wille, and I. Sengupta, "Improved mapping of quantum circuits to IBM QX architectures," *IEEE Trans. on CAD*, vol. 39, no. 10, pp. 2375–2383, 2020.

[18] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.