Reducing the Depth of Quantum Circuits Using Additional Circuit Lines

Nabila Abdessaied[1], Robert Wille[1,2], Mathias Soeken[1,2], and Rolf Drechsler[1,2]

[1] Institute of Computer Science, University of Bremen
Group of Computer Architecture, D-28359 Bremen, Germany
[2] Cyber-Physical Systems, DFKI GmbH
D-28359 Bremen, Germany
{nabila,rwille,msoeken,drechsle}@informatik.uni-bremen.de

Abstract. The synthesis of Boolean functions, as they are found in many quantum algorithms, is usually conducted in two steps. First, the function is realized in terms of a reversible circuit followed by a mapping into a corresponding quantum realization. During this process, the number of lines and the quantum costs of the resulting circuits have mainly been considered as optimization objectives thus far. However, beyond that also the depth of a quantum circuit is vital. Although first synthesis approaches that consider depth have recently been introduced, the majority of design methods did not consider this metric.

In this paper, we introduce an optimization approach aiming for the reduction of depth in the process of mapping a reversible circuit into a quantum circuit. For this purpose, we present an improved (local) mapping of single gates as well as a (global) optimization scheme considering the whole circuit. In both cases, we incorporate the idea of exploiting additional circuit lines which are used in order to split a chain of serial gates. Our optimization techniques enable a concurrent application of gates which significantly reduces the depth of the circuit. Experiments show that reductions of approx. 40% on average can be achieved when following this scheme.

1 Introduction

Quantum computation has become an active research field due to its promising results for important tasks such as factorization or database search. Motivated by this, researchers have developed several synthesis approaches [1–5]. Many quantum algorithms are often described by means of a structured quantum circuit in which only the representation of Boolean components differs. Hence, for the synthesis of these components into quantum circuits, usually a two-step approach is applied: First, the desired Boolean functionality is realized as a reversible circuit only consisting of reversible gates which is afterwards mapped to an equivalent realization based on quantum gates. For this purpose, mapping schemes as introduced e.g. in [6, 15] are applied.

In this flow, minimizing the number of lines and the quantum costs have been considered as the major optimization objectives thus far. However, beyond that

G.W. Dueck and D.M. Miller (Eds.): RC 2013, LNCS 7948, pp. 221–233, 2013.

also the depth of the circuit is vital. Depth optimization techniques consider the concurrent application of single gates in order to reduce the overall execution time of the circuit realization.

While first approaches for synthesis with respect to depth have recently been introduced (see e.g. [7–11]), the vast majority of design methods does not consider this metric. As an example in [7, 11], a cycle representation was chosen and input cycles where partitioned into three subsets. Each subset is synthesized independently on a different set of ancillae in parallel. This method requires $2n$ additional lines and focuses only on reducing the depth of reversible circuit rather than the quantum circuit. This is crucial since the execution times for two reversible gates can differ significantly when taking the respective quantum circuit mapping into account. As a consequence, even a depth-optimal reversible circuit likely leads to a quantum circuit with non-optimal depth. Another post-synthesis approach has been presented in [8]. However, their approach makes use of a special class of templates. Finally, the work presented in [10] describes an exhaustive algorithm aiming to find a minimal depth quantum circuit using a special gate library. However, due to its exponential time complexity, it is only applicable to circuits with a small number of qubits.

In this paper, we present an idea on how depth of quantum circuits can be reduced by adding an additional line to the circuit. Based on this idea, two depth optimization approaches are presented. The first method aims to reduce the depth by applying the reduction gate-per-gate, whereas the second method focuses on the whole circuit. An experimental evaluation of both approaches shows that a significant improvement of depth can be achieved for quantum circuits.

The remainder of this paper is structured as follows. The next section briefly introduces reversible and quantum circuits. Depth metrics and the general idea are presented in Sect. 3. Afterwards, both proposed approaches are described and evaluated in Sect. 4 and Sect. 5, respectively. Finally, Sect. 6 concludes the paper.

2 Background

To keep the remainder of this paper self-contained, this section briefly introduces the basics on reversible circuits, quantum circuits, and the corresponding mapping from reversible to quantum circuits.

2.1 Reversible Circuits

Boolean reversible functions are those functions $f : \mathbb{B}^n \to \mathbb{B}^n$ that are bijective, i.e. there exists an 1-to-1 mapping from the inputs to the outputs and vice versa. Reversible functions can be realized by reversible circuits that consist of at least n lines. Reversible circuits are cascades of reversible gates that belong to a gate library. One gate library that is often used consists of multiple control Toffoli gates [12].

(a) Reversible gate (b) Reversible circuit

Fig. 1. Reversible circuitry

Definition 1. *Given a set of variables* $\mathcal{V} = \{x_1, \ldots, x_n\}$*, a* multiple control Toffoli *gate* $\mathrm{T}(C, t)$ *has* control lines $C = \{x_{j_1}, x_{j_2}, \ldots, x_{j_l}\} \subset \mathcal{V}$ *and a* target line $t \in \mathcal{V} \setminus C$*. The gate maps* $t \mapsto t \oplus (x_{j_1} \wedge x_{j_2} \wedge \cdots \wedge x_{j_l})$ *and leaves all other lines unaltered. In the special cases* $|C| = 0$ *and* $|C| = |\{c\}| = 1$*, the gates are referred to as* NOT *and* CNOT *gate and denoted* $\mathrm{N}(t)$ *and* $\mathrm{C}(c, t)$*, respectively.*

In [13], it has been shown that any reversible function $f : \mathbb{B}^n \to \mathbb{B}^n$ can be realized by a reversible circuit with n lines when using Toffoli gates.

Example 1. Figure 1(a) shows a Toffoli gate with two control lines. The control lines are denoted by $\bullet$, while the target line is denoted by $\bigoplus$. The annotated values demonstrate the computation of the gate for a given input assignment. Figure 1(b) shows different Toffoli gates in a cascade forming a reversible circuit.

2.2 Quantum Boolean Circuits

Instead of bits, quantum circuits manipulate qubits which can represent the classical Boolean values but also the superposition of them. More precisely, a *qubit* $|\varphi\rangle$ is a vector $\binom{a}{b}$ where $a, b \in \mathbb{C}$ such that $|a|^2 + |b|^2 = 1$. If $a = 1$, then $|\varphi\rangle$ represents the classical 0, denoted $|0\rangle$, and if $b = 1$, then $|\varphi\rangle$ represents the classical 1, denoted $|1\rangle$.

In general, a quantum gate acting on n qubits represents a $2^n \times 2^n$ unitary matrix [14], where a matrix U is unitary if $U^\dagger U = UU^\dagger = I$ and $U^\dagger$ is the adjoint matrix $U^\dagger = U^{*^T}$. Using this gate definition, many quantum mechanical effects such as superposition and entanglement can be formulated. However, in the scope of this paper we are considering circuits that realize pure Boolean functionality but still need to be realized using quantum gates in order to embed them into quantum algorithms such as Deutsch-Josza, Grover, or Shor. Toffoli gates represent a unitary matrix and are hence suitable for realizing quantum Boolean circuits. However, with respect to the actual physical implementation, it is of interest to obtain circuits that make use of gates from a library with only a few elements [6]. For the present paper, we are making use of a common gate library consisting of four quantum gates that only change one qubit at a time and is defined as follows.

Definition 2. *A quantum gate* $U(C, t)$ *applies the unitary* 2×2 *matrix to the qubit that corresponds to the* target line t*, if and only if all* control lines C *are*

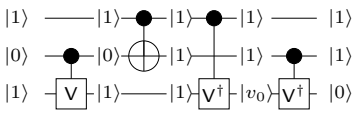
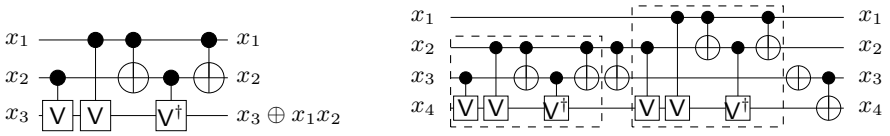


Fig. 2. Quantum circuitry



(a) Based on Fig. 1(a)

(b) Based on Fig. 1(b)

Fig. 3. Mapping reversible circuits to quantum circuits

assigned 1. *We consider a gate library* $X(\{\}, t)$, $X(\{c\}, t)$, $V(\{c\}, t)$, *and* $V^\dagger(\{c\}, t)$ *with* $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $VV = X$, *and* $V^\dagger$ *being the adjoint of* V.

Note that $X(\{\}, t) = \mathrm{N}(t)$ and $X(\{c\}, t) = \mathrm{C}(c, t)$. The gate library is often referred to as NCV library.

Example 2. Figure 2 depicts a quantum circuit consisting of four gates, where $|v_0\rangle = V^\dagger|1\rangle = V|0\rangle$.

2.3 Mapping Reversible Circuits to Quantum Circuits

Since any quantum operation can be represented by a unitary matrix [14], each quantum circuit is inherently reversible. As discussed above, when reversible circuits should be represented as a quantum circuit, the Toffoli gates are too general and, thus, not suitable for a realization. As a consequence, reversible circuits are mapped to quantum circuits that only consists of gates of a particular gate library, e.g. the NCV library. For this purpose, each gate of the reversible circuit is *mapped* into a cascade of functionally equivalent quantum gates.

Example 3. Consider a Toffoli gate with two control lines as shown in Fig. 1(a). A functionally equivalent realization in terms of quantum gates is depicted in Fig. 3(a). This cascade can be applied to fully map the reversible circuit shown in Fig. 1(b) into an equivalent quantum circuit. For this purpose, all corresponding Toffoli gates are respectively substituted with a corresponding quantum gate cascade. The 2^{nd}, 4^{th}, and 5^{th} gate remain unchanged as they already represent quantum gates. The resulting fully equivalent quantum circuit is shown in Fig. 3(b).

Similar mappings exist for Toffoli gates with more than two control lines. But with increasing number of control lines, the resulting quantum circuits become

more expensive, i.e. require more quantum gates. The currently best known mappings of single Toffoli gates into quantum cascades have been introduced in [15]. In this work, we are following the mappings introduced there. As single quantum gates are assumed to have unit costs, the number of gates of the resulting cascades usually is referred to as *quantum costs*.

3 Reducing the Depth of Quantum Circuits

In this work, we are proposing optimization approaches aiming for a reduction of the depth in quantum circuits using additional circuit lines. This section first motivates the consideration of depth in quantum circuits, whereas the general idea of the proposed approaches is outlined afterwards.

3.1 Consideration of Depth in Quantum Circuits

Thus far, the major optimization objectives for synthesis have been the number of lines and the quantum costs of the resulting circuits as reviewed above. However, beyond that also the *depth* of a quantum circuit is vital. This metric recognizes whether gates can concurrently be applied which likely leads to a reduction in the execution time of a circuit.

Definition 3. *Let $U_i(C_i, t_i)$ and $U_{i+1}(C_{i+1}, t_{j+1})$ be two consecutive quantum gates. These gates can be applied* concurrently *if*

$$|C_i \cup C_{i+1} \cup \{t_i, t_{i+1}\}| = |C_i| + |C_{i+1}| + 2.$$

In other words, if the lines used by each gate (both control and target line) are disjoint. Let G be a quantum circuit with k elementary quantum gates, then G can be partitioned into $m \leq k$ subcircuits whose gates can be pairwise applied concurrently. We refer to the minimal m as the depth *of the circuit.*

Algorithm D (*Determine Circuit Depth*). Given a quantum circuit $G = U_1(C_1, t_1) \dots U_k(C_k, t_k)$ over n variables $x_1, \dots, x_n$. This algorithm determines the depth m of the circuit according to Definition 3 by applying a greedy search to gates that can be executed in parallel. For the computation, we are making use of the integers $b_1, \dots, b_n$.

D1. [Initialize.] Set $m \leftarrow 1$, $i \leftarrow 1$, and $b_j \leftarrow 0$ for $1 \leq j \leq n$.

D2. [Terminate?] If $i > k$, terminate.

D3. [Apply gate.] For each $x_j \in C_i \cup \{t_i\}$, set $b_j \leftarrow b_j + 1$.

D4. [Gates do not overlap?] If there exists no $j \in \{1, \dots, n\}$ such that $b_j = 2$, set $i \leftarrow i + 1$ and goto Step D2.

D5. [Gates overlap.] For each $j \in \{1, \dots, n\}$, set $b_j \leftarrow 1$, if $x_j \in C_i \cup \{t_i\}$, otherwise set $b_j \leftarrow 0$; set $m \leftarrow m + 1$, $i \leftarrow i + 1$, and goto Step D2. ∎

Example 4. Figure 4 illustrates the depth for the reversible circuit shown in Fig. 1(b).

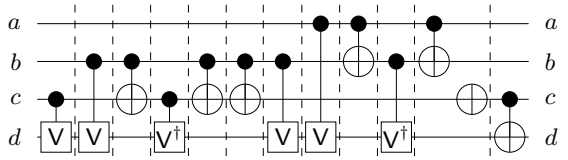


Fig. 4. Quantum depth for the reversible circuit shown in Fig. 1(b)

Although the *coherence time*, i.e. the time a qubit can keep its quantum state, and the *gate operation time*, i.e. the time a gate needs to perform its operation, may vary from one technology to another (see e.g. Table III of [16]), keeping the overall execution time as small as possible is essential in all these cases. Consequently, the depth metric can be applied in a generic manner, as it provides a proper model which can be considered already at the synthesis stage in the absence of precise technological constraints. Despite the fact that quantum algorithms already exploit algorithmic parallelism to increase the processing speed, synthesis approaches should aim for producing circuits with at least as possible circuit depth.

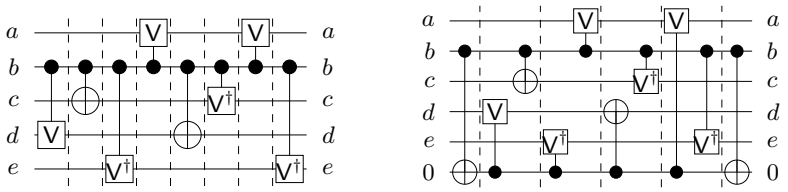
Motivated by this, we are considering the question how the depth of a quantum circuit can be reduced in the remainder of this paper. For this purpose, we are making use of additional circuit lines as motivated in the following.

3.2 Exploiting Additional Circuit Lines

Keeping the number of circuit lines as small as possible is well accepted in the synthesis of quantum circuits. This is mainly motivated by the fact that each circuit line has to be represented by a qubit, which is a very limited resource. Nevertheless, evaluations also showed that a (slight) extension of a circuit with additional lines may have significant benefits. For example in [6, 15], it has been demonstrated that a larger amount of circuit lines allow for a much cheaper mapping of reversible circuits to quantum circuits in terms of gate count. In [3], evaluations showed that using twice the number of circuit lines reduces the quantum costs by up to two orders of magnitude. Eventually, this led to a post-synthesis optimization approach [17] which enables reductions in quantum costs of up to 69% only by adding a single additional line to the circuit.

In this work, we show that similar concepts also help in reducing the depth of quantum circuits. We are following the established synthesis flow reviewed in Sect. 2.3, i.e. first a reversible circuit is realized which afterwards is mapped to a quantum circuit. However, by incorporating additional lines during this process, a depth-aware optimization becomes possible. The additional circuit lines are introduced as *helper lines*.

Definition 4. *Let G be a reversible or quantum circuit. A* helper line *is an additional line whose input is set to a constant* 0 *and is used in a way throughout the circuit such that the output of the line is also* 0.



(a) Initial circuit (b) Circuit with reduced depth

Fig. 5. Depth reduction by using additional helper lines

Following the concept from [17], helper lines can now be applied in order to "buffer" values of circuit lines so that they can be re-used later by other gates. Whenever the current value of a helper line h is 0, another signal line x can be copied to h by appending a *copy gate* $\mathrm{C}(\{x\}, h)$ to the circuit. The helper line can be restored with the same gate if no other gate has used h as target line in between.

In [17], this buffering has been exploited to remove common control lines connections between Toffoli gates in order to reduce the quantum cost. However, the same concept can similarly be applied to reduce the depth of quantum circuits as illustrated by the following example.

Example 5. Figure 5(a) shows a circuit in which no gates can be performed in parallel since they all share the same control line b. In Fig. 5(b) a helper line has been added to copy the value of b. By doing this, the gates can be rearranged which reduces the depth from 8 to 6.

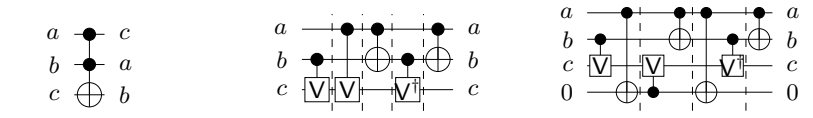
Clearly, Example 5 presents a rather artificial circuit. However, based on this general idea we are proposing different optimization approaches whose evaluations show that indeed a significant reduction of depth in quantum circuits can be achieved.

4 Optimization Approaches

Motivated by the general idea outlined above, two optimization approaches are proposed in this section which aim for reducing the depth by exploiting additional circuit lines. The first approach follows a local scheme, i.e. considers each Toffoli gate independently, where the second approach considers the whole circuit instead. Finally, techniques are presented to further reduce the depth and the quantum costs which can be applied to the resulting quantum cascades.

4.1 Consideration of Single Toffoli Gates

The availability of a helper line as introduced in the previous section allows for an improvement of the mapping scheme reviewed in Sect. 2.3. Recall that, when



(a) Toffoli Gate (b) Original mapping (c) Proposed mapping

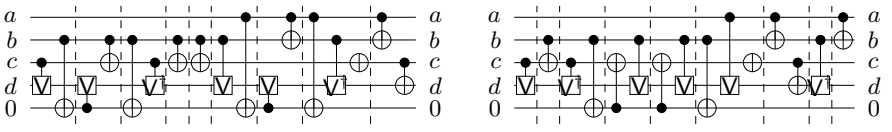
Fig. 6. Consideration of single Toffoli gates

following the default mapping scheme, each Toffoli gate is mapped to a quantum realization of depth 5 as shown in Fig. 6(b). However, as the second and the third gate share the same control line, an additional helper line allows for a concurrent execution of both gates as shown in Fig. 6(c). Since additionally the copy gates can be inserted without increasing the depth, a depth reduction for the quantum circuit realization for each Toffoli gate from 5 to 4 can be obtained.

Example 6. Consider again the reversible circuit from Fig. 1(b). Using the established mapping scheme from Sect. 2.3, a quantum circuit with depth 12 results (as shown in Fig. 3(b); none of the gates except for the single NOT gate can be executed concurrently). In contrast, applying the additional helper line as proposed in Fig. 6, the circuit depicted in Fig. 7(a) results. This reduces the depth from 12 to 9.

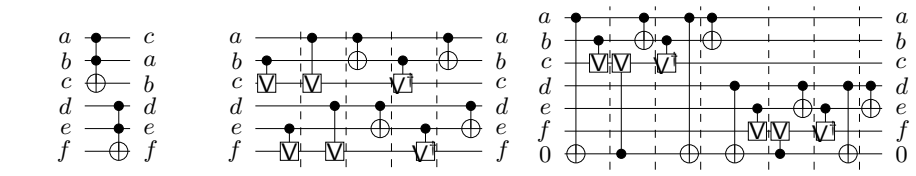
Note that this procedure can also be applied to Toffoli gates with more than two control lines. In fact, state-of-the-art mapping schemes (such as described in [15]) decompose these gates into cascades of two-controlled Toffoli gates. For them, the depth-optimized mapping to quantum gates as proposed in Fig. 6 can be applied. Moreover, the same scheme can be applied to other reversible gates such as the Peres gate as well.

This scheme is not beneficial in all cases. In fact, if concurrent Toffoli gates are mapped to a quantum circuit, the original mapping leads to better results. This is illustrated by means of Fig. 8. Applying the original mapping scheme to the two Toffoli gates shown in Fig. 8(a) leads to the quantum cascade as shown in Fig. 8(b). As both Toffoli gates are applied concurrently, also the resulting quantum gate cascades can be applied concurrently, i.e. a depth of 5 results. Applying the proposed scheme from Fig. 6 would worsen the result. In fact, the helper line together with the required copy gates would increase the depth to 7 as shown in Fig. 8(c).



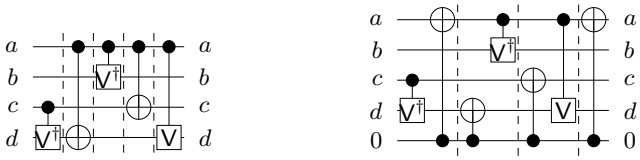
(a) Consideration of single Toffoli gates (b) Consideration of the whole circuit

Fig. 7. Application of the proposed approaches to the circuit from Fig. 1(b)



(a) Original circuit (b) Original mapping (c) Proposed mapping

Fig. 8. Application of the local scheme to concurrent Toffoli gates



(a) Original circuit (b) Resulting circuit

Fig. 9. Consideration of the whole circuit

Consequently, this scheme is only applied in cases where an actual depth improvement can be achieved. However, experiments summarized in Sect. 5 clearly confirm that substantial improvements with respect to the depth can still be achieved. As a drawback, this obviously comes with the price of increased quantum costs in the resulting cascade. But also here, experiments show the resulting increase to be moderate.

4.2 Consideration of the Whole Circuit

While so far the helper line has been exploited in a local context, also a global consideration turns out to be beneficial. The idea is to identify subcircuits of gates sharing the same control line and use the helper line in order to partition the gates. Then, each consecutive pair of gates in such a cascade can concurrently be executed by using the original control line for the first gate and the copied value at the helper line for the second gate.

Example 7. Figure 9(a) shows a quantum circuit composed of gates that share the same control lines. Using the helper line, an equivalent realization as shown in Fig. 9(b) can be derived. This reduces the depth from 5 to 4.

This scheme can additionally be improved by applying the *moving rule* for quantum circuits. In fact, two adjacent gates $U(C_1, t_1)$ and $U(C_2, t_2)$ can be interchanged if $t_2 \notin C_1$ and $t_1 \notin C_2 \cap \{t_1\} = \emptyset$. As a result, gates can be moved through the circuit which might lead to larger subcircuits of gates sharing the same control line. In this case, a more substantial reduction can be achieved.

Example 8. Consider again the quantum circuit shown in Fig. 4. The second, fifth, sixth, and seventh gate share the same control line b *and* can be moved together (note, although also the third and tenth gate have control line b, they cannot be moved to a consecutive cascade). Exploiting that, this cascade can be optimized leading to the circuit shown in Fig. 7(b). This reduces the depth from 12 to 9.

Note that this scheme also increases the quantum costs of the resulting circuit. However, since for each identified subcircuit only two copy gates have to be added, the increase is almost negligible.

4.3 Further Optimizations

Independent of the optimization schemes proposed above, the depth of quantum gate cascades can additionally be improved using existing optimization schemes that originally aimed for quantum cost reduction. In particular, the application of merging and deletion rules as explained in [18] together with the moving rule as already discussed above is beneficial. For example, the circuit shown in Fig. 7(b) (obtained using existing mapping schemes) obviously can be improved by removing the fifth and the sixth gate which cancel each other. This reduces the quantum costs but also improves the depth of the circuit. Accordingly, such simple optimizations are also applied in our approach. For the experimental evaluation summarized in Sect. 5, the methods exploiting additional helper lines are applied to circuits already optimized using moving, merging, and deletion rule.

5 Experimental Results

In order to confirm the efficiency of the proposed idea, the approaches described above have been implemented and experimentally evaluated. For this purpose, the open source toolkit *RevKit* [19] has been applied and benchmarks have been taken from the *RevLib* [20] database. All experiments have been conducted on an Intel Core i5 Processor with 4 GB of main memory. In this section, we summarize and discuss the obtained results.

Table 1 provides the obtained numbers. For all benchmarks listed in the first column, the number of lines (*Lines*), the quantum costs (*Costs*), and the depth (*Depth*) of the respective circuit realizations as well as the run-time (*Time*) needed to generate them are provided. We distinguish between the following circuits:

- Initial Circuits (IC) represent the circuits as taken from *RevLib* and mapped to quantum circuits as described in Sect. 2.3, i.e. without any depth optimization whatsoever.
- Optimized Circuits (OC) represent the circuits that have additionally been optimized using the straightforward techniques reviewed in Sect. 4.3.

Table 1. Experimental evaluation

Benchmark	Initial Circuits				Optimized Circuits (Sect. 4.3, +0 line)				Local (Sect. 4.1, +1 line)					Global (Sect. 4.2, +1 line)				
	Lines	Cost	Depth	Time	Cost	Depth	Time	$Impr_{IC}$	Cost	Depth	Time	$Impr_{IC}$	$Impr_{OC}$	Cost	Depth	Time	$Impr_{IC}$	$Impr_{OC}$
ex5p_296	206	1843	1584	0,00	1578	1352	46,95	15%	1775	303	181,48	81%	78%	1578	226	139,92	86%	83%
hwb9_304	170	2275	1916	0,01	1853	1624	77,71	15%	2144	424	410,64	78%	74%	1853	305	313,58	84%	81%
c2_181	35	368	280	0,00	344	262	0,04	6%	378	71	0,73	75%	73%	344	71	0,66	75%	73%
hwb8_303	112	1461	1226	0,01	1202	1049	20,88	14%	1410	294	116,59	76%	72%	1202	212	89,21	83%	80%
hwb7_302	73	909	769	0,01	754	658	4,09	14%	898	222	34,28	71%	66%	754	167	28,44	78%	75%
bw_291	87	943	788	0,01	782	681	5,26	14%	932	230	29,11	71%	66%	782	189	21,47	76%	72%
hwb6_301	46	507	426	0,00	432	376	0,67	12%	507	136	5,79	68%	64%	432	109	4,42	74%	71%
add64_184	193	768	642	0,00	642	516	0,23	20%	770	197	10,06	69%	62%	642	198	5,99	69%	62%
add32_183	97	384	322	0,00	322	260	0,06	19%	386	101	1,35	69%	61%	322	102	0,85	68%	61%
add16_174	49	192	162	0,00	162	132	0,02	19%	194	53	0,19	67%	60%	162	54	0,14	67%	59%
ham15_298	45	309	279	0,00	242	209	0,51	25%	282	84	1,35	70%	60%	242	77	1,22	72%	63%
add8_172	25	96	82	0,00	82	68	0,00	17%	98	29	0,03	65%	57%	82	30	0,03	63%	56%
hwb5_300	28	276	237	0,00	237	208	0,11	12%	289	92	1,15	61%	56%	237	83	0,95	65%	60%
mod5adder_306	32	292	252	0,00	249	219	0,17	13%	292	97	1,80	62%	56%	249	77	1,55	69%	65%
rd73_312	25	217	186	0,00	180	158	0,08	15%	206	71	0,80	62%	55%	182	63	0,73	66%	60%
rd84_313	34	304	258	0,00	247	218	0,30	16%	295	99	2,65	62%	55%	247	71	1,79	72%	67%
e64-bdd_295	195	907	748	0,01	779	620	3,00	17%	973	322	58,58	57%	48%	779	323	57,99	57%	48%
cnt3-5_179	16	65	56	0,00	62	53	0,00	5%	71	28	0,03	50%	47%	62	34	0,04	39%	36%
ham7_299	21	141	124	0,00	111	96	0,03	23%	144	56	0,14	55%	42%	111	49	0,12	60%	49%
mod5d2_70	5	16	14	0,00	15	13	0,00	7%	18	10	0,00	29%	23%	15	9	0,00	36%	31%
urf2_277	8	7521	6750	0,04	7172	6390	10,19	5%	7885	4998	347,74	26%	22%	7244	4832	330,03	28%	24%
4mod5-v0_18	5	25	24	0,00	16	14	0,00	42%	22	11	0,00	54%	21%	16	9	0,00	63%	36%
aryy6_256	17	5593	5491	0,03	4873	4810	20,36	12%	5197	3963	517,14	28%	18%	4949	3877	647,55	29%	19%
alu3_200	18	3387	3322	0,01	2970	2926	8,19	12%	3170	2413	262,46	27%	18%	2982	2363	319,25	29%	19%
Average Improvement								15%				60%	52%				63%	56%

Both, the initial circuits and optimized circuits, allow for a comparison to the circuits obtained by the proposed techniques, namely:

- Circuits that have been obtained by using the optimization scheme that considers single Toffoli gates (LOCAL) as described in Sect. 4.1.
- Circuits that have been obtained by using the optimization scheme that considers the whole circuit (GLOBAL) as described in Sect. 4.2.

The percentage depth-improvement of the circuits obtained by the proposed techniques with respect to the initial circuit and the optimized circuits are provided in the columns denoted by $Impr_{IC}$ and $Impr_{OC}$, respectively.

First of all, it can be observed that already the naive approaches reviewed in Sect. 4.3 lead to significant improvements (15% on average and up to 42% in the best case for *4mod5-v0_18*). However, exploiting additional circuit lines enables further improvements which are factors beyond that. In the best case (*ex5p_296*), depth can be reduced from 1352 to 303 (using the local approach from Sect. 4.1) or 226 (using the global approach from Sect. 4.2). But also for the other benchmarks substantial reductions can be observed, even compared to the already optimized circuits.

As discussed above, these improvements in the depth may come at the price of higher quantum costs. As our evaluations show, this particularly holds for the local consideration of single Toffoli gates (see columns denoted LOCAL). Here, quantum costs increase by 18% on average compared to the already optimized circuit. However, for the global scheme, no such disadvantages can be observed. In fact, quantum costs remain unchanged here (see columns denoted GLOBAL).

Overall, even compared to already optimized circuits, improvements of more than 50% on average can be achieved. If the global scheme is applied, these achievements are possible without the need to accept an increase in the quantum costs. This is made possible by the addition of a single circuit line. Although this eventually results in the consideration of another qubit to be physically realized, the possible benefits with respect to timing and particularly decoherence time might be worth the overhead.

6 Conclusion

In this paper, depth optimization by adding a helper line to quantum circuits has been introduced and evaluated. Two approaches, namely gate based and circuit based, have been considered. Experimental results for the two methods have shown significant depth reductions which reaches over 50% for quantum circuits. Although these methods increase quantum cost, applying further improvements to the quantum circuits have fixed the problem.

References

1. Fazel, K., Thornton, M., Rice, J.: ESOP-based toffoli gate cascade generation. In: Pacific Rim Conference on Communications, Computers and Signal Processing, PacRim 2007, pp. 206–209 (2007)

2. Miller, D.M., Maslov, D., Dueck, G.W.: A transformation based algorithm for reversible logic synthesis. In: Design Automation Conf., pp. 318–323 (2003)
3. Wille, R., Drechsler, R.: BDD-based synthesis of reversible logic for large functions. In: Design Automation Conf., pp. 270–275 (2009)
4. Soeken, M., Wille, R., Otterstedt, C., Drechsler, R.: A synthesis flow for sequential reversible circuits. In: Int'l Symposium on Multiple-Valued Logic, pp. 299–304 (2012)
5. Soeken, M., Wille, R., Hilken, C., Przigoda, N., Drechsler, R.: Synthesis of reversible circuits with minimal lines for large functions. In: ASP Design Automation Conf., pp. 85–92 (2012)
6. Barenco, A., Bennett, C.H., Cleve, R., DiVinchenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. The American Physical Society 52, 3457–3467 (1995)
7. Arabzadeh, M., Saheb Zamani, M., Sedighi, M., Saeedi, M.: Depth-optimized reversible circuit synthesis. Quantum Information Processing, 1–23 (2012)
8. Maslov, D., Dueck, G., Miller, D., Negrevergne, C.: Quantum circuit simplification and level compaction. Transactions on Computer-Aided Design of Integrated Circuits and Systems 27(3), 436–444 (2008)
9. Bocharov, A., Svore, K.M.: A depth-optimal canonical form for single-qubit quantum circuits arXiv preprint arXiv:1206.3223 (2012)
10. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits arXiv preprint arXiv:1206.0758 (2012)
11. Arabzadeh, M., Zamani, M., Sedighi, M., Saeedi, M.: Logical-depth-oriented reversible logic synthesis. In: Int'l Workshop on Logic and Synthesis (2011)
12. Toffoli, T.: Reversible computing. In: de Bakker, J., van Leeuwen, J. (eds.) Automata, Languages and Programming. LNCS, vol. 85, pp. 632–644. Springer, Heidelberg (1980)
13. Shende, V.V., Prasad, A.K., Markov, I.L., Hayes, J.P.: Synthesis of reversible logic circuits. Transactions on Computer-Aided Design of Integrated Circuits and Systems 22(6), 710–722 (2003)
14. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge Univ. Press (2000)
15. Miller, D.M., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control Toffolli gates. In: Int'l Symp. on Multi-Valued Logic, pp. 217–222 (May 2011)
16. Meter, R.V., Oskin, M.: Architectural implications of quantum computing technologies. J. Emerg. Technol. Comput. Syst. 2(1), 31–63 (2006)
17. Miller, D.M., Wille, R., Drechsler, R.: Reducing reversible circuit cost by adding lines. In: Int'l Symp. on Multi-Valued Logic, pp. 217–222 (2010)
18. Zahra, S.: Technology Mapping and Optimization for Reversible and Quantum. PhD thesis, University of Victoria (2012)
19. Soeken, M., Frehse, S., Wille, R., Drechsler, R.: Revkit: an open source toolkit for the design of reversible circuits. In: De Vos, A., Wille, R. (eds.) RC 2011. LNCS, vol. 7165, pp. 64–76. Springer, Heidelberg (2012), RevKit is available at `http://www.revkit.org`
20. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: an online resource for reversible functions and reversible circuits. In: Int'l Symp. on Multi-Valued Logic, pp. 220–225 (2008), RevLib is available at `http://www.revlib.org`