Constant-Factor Optimization of Quantum Adders on 2D Quantum Architectures

Mehdi Saeedi, Alireza Shafaei, Massoud Pedram

Department of Electrical Engineering, University of Southern California,
Los Angeles, CA 90089-2562
{msaeedi, shafaeib, pedram}@usc.edu

Abstract. Quantum arithmetic circuits have practical applications in various quantum algorithms. In this paper, we address quantum addition on 2-dimensional nearest-neighbor architectures based on the work presented by Choi and Van Meter (JETC 2012). To this end, we propose new circuit structures for some basic blocks in the adder, and reduce communication overhead by concurrent optimization of consecutive blocks and also by parallel execution of expensive Toffoli gates. The proposed optimizations reduce total depth from $140\sqrt{n}+k_1$ to $92\sqrt{n}+k_2$ for constants k_1, k_2 and affect the computation fidelity considerably.

Keywords: Quantum Adders, 2D Quantum Architectures, Nearest Neighbor Interaction.

1 Introduction

Quantum algorithms are often described in the quantum circuit model of computation, where for a quantum circuit with n qubits, any pairs of qubits can interact. However, current advances in physical quantum technologies can allow qubit interactions in one-, two-, or three-dimensional spaces. Restricting interactions to only linear dimension results in $O(n)$ overhead. On the other hand, working with 2D (or 3D) quantum architectures where each qubit can interact with 4 (or 6) neighboring qubits provides more flexibility.

For a given quantum circuit C one can construct an interaction graph $G_C = (V_C, E_C)$, the nodes of which represent qubits in C with edges between them when a gate in C involves the related qubits. Additionally, the architecture (or fabric) of a quantum computing system can be described by a simple connected graph $G_Q = (V_Q, E_Q)$ where vertices V_Q represent qubits and edges E_Q represent adjacent qubit pairs that gates can be applied on [1]. Accordingly, the problem of mapping a quantum circuit C with arbitrary interactions between qubits onto a quantum architecture with limited interaction distance can be mapped to the problem of embedding graph G_C into graph G_Q.

In general, the graph embedding problem is NP-hard. However, optimal embedding methods with polynomial time complexities for several classes of graphs have been proposed [2]. In [3], the concept of *dilation* in graph embedding has been applied to find a depth lower bound for a quantum circuit after embedding.

In this case, dilation is defined as the maximum distance between adjacent nodes of the graph after embedding. Working with proven properties of log-depth binary tree and considering the fact that log-depth quantum addition circuits exist, Choi and Van Meter [3] showed that the depth lower bound of the exact quantum addition circuit on a k-dimensional quantum architecture is $\Omega(\sqrt[k]{n})$. In [4], the authors examined the minimum overhead in depth for emulating a circuit C by a circuit C' subject to the constraints imposed by the interaction constraints and showed that this overhead is $O(n)$ for 1D, $O(\sqrt{n})$ for 2D, $O(\log^2 n)$ or $O(\log n)$ (depending on the approach) for hypercube.

Exploring an efficient realization of a given quantum algorithm or quantum circuit for a restricted architecture has been followed by a number of researchers during the recent years. Physical implementations of the quantum Fourier transform (QFT) [5, 6], Shor's factorization algorithm [7–9], quantum error correction [10], and general reversible circuits [11] for 1D/2D architectures have been explored in the past. Worst-case synthesis cost of a general/Boolean unitary matrix under the 1D restriction has been discussed in [12–15]. In [16, 17] heuristic methods for converting an arbitrary quantum circuit to its equivalent circuit on 1D architectures have been proposed.

Quantum adder and its modular version have applications in different quantum algorithms including Shor's factoring algorithm. In [18], a quantum adder with $\Theta(\sqrt{n})$ depth on 2D quantum architectures was proposed which has $140\sqrt{n}-72$ depth, in terms of one- and two-qubit quantum gates. Asymptotically, the depth of the proposed adder is optimal. However, constant-factor optimization is possible and in fact desirable. Besides the effect of reducing circuit size/depth on physical realization, any additional gate in the circuit longest path can reduce circuit fidelity to some extent. Based on the analysis done in [19] for fault-tolerant error correction with a concatenated 7-qubit CSS code [20], nearest-neighbour communication overhead results in 175x reduction in error threshold. Improving error threshold is costly and may include using a more sophisticated quantum control protocol to have gates with higher fidelities or applying a more robust error correction code. Therefore, reducing unnecessary communication overhead for a useful quantum computation is vital. Considering the effect of addition on modular multiplication and modular exponentiation circuits and recent papers [9, 21, 22], reducing communication overhead for quantum adder by circuit optimization — the focus of this work — is of particular interest.

In this paper, we show how $140\sqrt{n}$+const depth in [18] can be further improved to $92\sqrt{n}$+const. For this purpose, we reconsider the basic blocks in the suggested quantum adder and introduce some constant-factor optimizations in communication overhead in different stages. To physically implement a given circuit, one needs to decompose all gates into primitive one- and two-qubit gates. To decompose a 3-qubit Toffoli ($\mathcal{T}$) gate, we use Clifford+T gates which are universal and have fault-tolerant (FT) implementation [20]. Fig. 1 shows the decomposition of the Toffoli gate into one- and two-qubit gates. To consider depth, we report circuit depth in terms of single-qubit, CNOT ($\mathcal{C}$) and SWAP ($\mathcal{S}$) gates. The rest of this paper is organized as follows. In Section 2, the method in [18] is

discussed. We introduce the reduction techniques in Section 3. The result of the proposed reductions is analyzed in Section 4 and Section 5. We finally conclude the paper in Section 6.

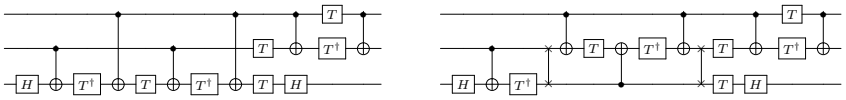


Fig. 1. Decomposition of the Toffoli gate into one-qubit and six CNOT gates [23] and the implementation with adjacent qubits.

2 Quantum Addition on 2D Architectures

In this section, we describe the circuit structure in [18] for quantum addition on 2D architectures. For an n-qubit quantum circuit, the method in [18] arranges the qubits in $\sqrt{n} \times \sqrt{n}$ array where each qubit can interact with its four neighboring qubits with no cost. Additionally, the circuit was divided into 3 phases which are executed sequentially. In the first phase, ripple-carry addition is performed on the first column, and carry-lookahead addition is performed on the other $\sqrt{n}-1$ columns. In the second phase, carry propagation is performed between columns, and finally in phase 3 carry generation and summation are performed.

In the first phase, after using a half-adder and $\sqrt{n}-1$ full-adders output carries $c_2, \cdots c_{\sqrt{n}+1}$ will be available. It is done in $32\sqrt{n}-17$ unit-time steps in [18]. The carry-lookahead addition in other columns produces

$$g_{k\sqrt{n}+j} = a_{k\sqrt{n}+j} \cdot b_{k\sqrt{n}+j} \tag{1}$$

$$p_{k\sqrt{n}+j} = a_{k\sqrt{n}+j} \oplus b_{k\sqrt{n}+j} \tag{2}$$

for $1 \leq k \leq \sqrt{n}-1$ and $1 \leq j \leq \sqrt{n}$. After computing g_i and p_i values in all columns in parallel, $G[i,j]$ and $P[i,j]$ are computed in serial based on (3) and (4) for $1 \leq k \leq \sqrt{n}-1$, and $2 \leq j \leq \sqrt{n}$ where $G[k\sqrt{n}+1, k\sqrt{n}+1] = g_{k\sqrt{n}+1}$ and $P[k\sqrt{n}+1, k\sqrt{n}+1] = p_{k\sqrt{n}+1}$. This part takes $34\sqrt{n}-19$ time steps in [18]. Accordingly, the first phase in [18] results in $34\sqrt{n}-19$ time steps.

$$G[k\sqrt{n}+1, k\sqrt{n}+j] = g_{k\sqrt{n}+j} \oplus p_{k\sqrt{n}+j} \cdot G[k\sqrt{n}+1, k\sqrt{n}+j-1] \tag{3}$$

$$P[k\sqrt{n}+1, k\sqrt{n}+j] = p_{k\sqrt{n}+j} \cdot P[k\sqrt{n}+1, k\sqrt{n}+j-1] \tag{4}$$

In the second phase, column-level carries are computed as shown in (5) for $1 \leq k \leq \sqrt{n}-1$ in $18\sqrt{n}-18$ time steps.

$$c_{(k+1)\sqrt{n}+1} = G[k\sqrt{n}+1, (k+1)\sqrt{n}] \oplus c_{k\sqrt{n}+1} \cdot P[k\sqrt{n}+1, (k+1)\sqrt{n}] \tag{5}$$

In phase 3 output carries are calculated sequentially as (6) for $1 \leq k \leq \sqrt{n}-1$ and $j = \sqrt{n}-1, ..., 1$.

Table 1. Basic blocks in 2D adder [18] and their depths in terms of unit-cost gates. The last term (i.e., 3) in total depth represents 2 NOTs and one CNOT gate used to construct the final output in [18].

Name	#steps: gate sequence	Circuit
H, T, CNOT (C), SWAP ($\mathcal{S}$)	1	
Toffoli ($\mathcal{T}$(a,b,0))	14: 2 $\mathcal{S}$+ 12 1-qubit	H(0)C(b,0)T†(0)$\mathcal{S}$(b,0)C(a,b)T(b)C(0,b) T†(b)C(a,b)$\mathcal{S}$(b,0)T(b)T(0)C(a,b)H(0) T(a)T†(b)C(a,b)
Half-adder(a,b,0)	15: 1 $\mathcal{T}$+ 1 $\mathcal{C}$	$\mathcal{T}$(a,b,0)$\mathcal{T}$(a,b)
Full-adder(c,a,b,0)	32: 2 $\mathcal{T}$+ 2 $\mathcal{C}$+ 2 $\mathcal{S}$	$\mathcal{T}$(a,b,0)$\mathcal{T}$(a,b)$\mathcal{S}$(c,a)$\mathcal{T}$(a,b,0)$\mathcal{T}$(a,b)$\mathcal{S}$(c,a)
g,p(a,b,0)	15: 1 $\mathcal{T}$+ 1 $\mathcal{C}$	$\mathcal{T}$(a,b,0)$\mathcal{T}$(a,b)
G,P(P,G,a,p,g,0)	34: 2 $\mathcal{T}$+ 6 $\mathcal{S}$	$\mathcal{S}$(G,a)$\mathcal{S}$(P,G)T(a,p,g)$\mathcal{S}$(G,a)$\mathcal{S}$(g,0)T(a,p,g) $\mathcal{S}$(G,a)$\mathcal{S}$(P,G)$\mathcal{S}$(G,a)
Column_carry(P,G,C)	18: 1 $\mathcal{T}$+ 4 $\mathcal{S}$	$\mathcal{S}$(P,G)$\mathcal{T}$(C,G,P)$\mathcal{S}$(G,C)$\mathcal{S}$(P,G)$\mathcal{S}$(G,C)
Carry(P,G,a,p,C)	18: 1 $\mathcal{T}$+ 4 $\mathcal{S}$	$\mathcal{S}$(P,G)$\mathcal{S}$(p,C)$\mathcal{S}$(a,p)$\mathcal{T}$(a,G,P)$\mathcal{S}$(G,a)$\mathcal{S}$(P,G)
Carry1(p,g,c)	16: 1 $\mathcal{T}$+ 2 $\mathcal{S}$	$\mathcal{S}$(g,c)$\mathcal{T}$(p,g,c)$\mathcal{S}$(p,g)
SUM(c,P,a,p)	5 : 1 $\mathcal{C}$+ 4 $\mathcal{S}$	$\mathcal{S}$(c,P)$\mathcal{S}$(P,a)$\mathcal{T}$(a,p)$\mathcal{S}$(P,a)$\mathcal{S}$(c,P)
SUM1(c,a,p)	3 : 1 $\mathcal{C}$+ 2 $\mathcal{S}$	$\mathcal{S}$(c,a)$\mathcal{T}$(a,p)$\mathcal{S}$(c,a)
SUM2(p,c)	1 : 1 $\mathcal{C}$	$\mathcal{T}$(c,p)
phase 1	$34\sqrt{n}-19$: g,p + $(\sqrt{n}-1)$G,P	
phase 2	$18\sqrt{n}-18$: $(\sqrt{n}-1)$ Column_carry	
phase 3	$18\sqrt{n}+1$: $(\sqrt{n}-1)$ Carry + Carry1 + SUM1	
clearing ancillae	$70\sqrt{n}-39$: phase 1 + phase 2 + phase 3 - SUM1	
total depth	$140\sqrt{n}-72$: phase 1 + phase 2 + phase 3 + clearing ancillae + 3	

$$c_{k\sqrt{n}+j+1} = G[k\sqrt{n}+1, k\sqrt{n}+j] \oplus c_{k\sqrt{n}+1} \cdot P[k\sqrt{n}+1, k\sqrt{n}+j] \quad (6)$$

Finally, addition outputs are calculated as shown in (7) for $1 \le k \le \sqrt{n}-1$ and $1 \le j \le \sqrt{n}$. Altogether, operations in phase 3 can be performed in $18\sqrt{n}+1$ time steps.

$$s_{k\sqrt{n}+j} = a_{k\sqrt{n}+j} \oplus b_{k\sqrt{n}+j} \oplus c_{k\sqrt{n}+j} \quad (7)$$

Considering the three subcircuits for phase 1, phase 2, and phase 3 in sequence leads to $70\sqrt{n}-36$ time steps in [18]. Applying the inverse circuit to clear ancillae leads to $140\sqrt{n}-72$ time steps for the complete adder.

Based on the equations (1)-(7), Table 1 reports circuit depth in different blocks. In this table, we used the same notation in [18] for circuit blocks — g,p to compute g_i, p_i values in (1) and (2); G,P to compute $G[i,j]$ and $P[i,j]$ values in (3) and (4); Column_carry to compute column-level carries in (5); Carry & Carry1 to compute carries in (6); and SUM, SUM1 & SUM2 to compute final outputs in (7).

3 The Proposed 2D Adder

In this section, we revise the basic blocks in [18] and introduce additional parallelism in various parts to reduce circuit depth. Basically, the proposed optimizations are based on (1) new circuit structures for CARRY and SUM basic blocks (2) reducing communication overhead in Column_carry, (3) parallel execution of expensive Toffoli gates in G,P blocks as well as in Full-adders, and (4) reducing interaction overhead by concurrent optimization of two consecutive blocks.

3.1 New Circuits

Working with the same circuit structures in [18] for Half-adder, g,p, and G,P blocks as reported in Table 1, we define several new structures for the other blocks.

- **Full-adder**: The first $\mathcal{T}$ and $\mathcal{C}$ gates in the Full-adder blocks in [18] can be executed in parallel with the gates in the Half-adder circuit. This saves one $\mathcal{T}$ and one $\mathcal{C}$ for all $\sqrt{n}-1$ Full-adders.
- **Column_Carry**: Fig. 4 shows the new structure of Column_Carry block. In this circuit, $c[k\sqrt{n}+1]$ is from the previous column (e.g., c_4 in Fig. 2). After the computation, the new carry, e.g., c_7, is moved down, to be used by the next Column_Carry block. The previous carry, e.g., c_4 is placed near to the Carry module. This new structure saves 1 SWAP gate.
- **Carry**: Fig. 5 shows the new structure for Carry block. Since $c[k\sqrt{n}+1]$ is required to compute all carries in different rows, $c[k\sqrt{n}+1]$ is moved up in this figure. On the other hand, the generated carry is required to compute sum values, and hence is moved down. This new circuit uses 5 SWAP gates (vs. 4 in [18]).
- **SUM**: Applying the proposed circuit for Carry results in adjacent $c[k\sqrt{n}+j+1]$ and $p[k\sqrt{n}+j+1]$ values (see Fig. 5). Based on (7) sum outputs can be computed by a single CNOT gate. This saves 4 SWAP gates in [18]. In order to construct s_i values on b_i qubits, one needs to add one SWAP gate $\mathcal{S}(p[k\sqrt{n}+1], c[k\sqrt{n}+1])$. However, this SWAP gate can be removed because of an identical SWAP gate in the Carry circuit. Accordingly, we define another circuit block Carry1 with excluding the SWAP on $c[k\sqrt{n}+1]$ and $P[k\sqrt{n}+1][k\sqrt{n}+j]$ (for $j=1$) qubits. We do not need to use SUM1 and SUM2 blocks in the proposed 2D adder structure.

3.2 Reducing Communication Overhead

To use adjacent gates in the 2D quantum adder, we use a set of SWAP gates inside each circuit block. The added SWAP gates are used for communication between those gates required for the computation. In other words, the added SWAP gates are not required for the computation, and should be reduced as much as possible. Independent optimization of different blocks can reduce communication overhead inside each subcircuit, but has no view about the neighboring subcircuits. In this section, we consider consecutive circuit blocks to reduce communication overhead further. Note that the optimizations given in this section are based on the new circuit blocks given in Section 3.1.

- **G,P $\Rrightarrow$ Carry**: Reconsider (3), (4), and (6) and note that the result of Column_carry in (5), i.e., $c[k\sqrt{n}+1]$, is constructed on the last qubit in the Carry block (see Fig. 4 and Fig. 5). Fig. 6 shows the blocks in sequence. To simplify the circuit, note that the last three SWAP gates in G,P can be moved to right. Next, the resulting circuit can be reconstructed as shown in

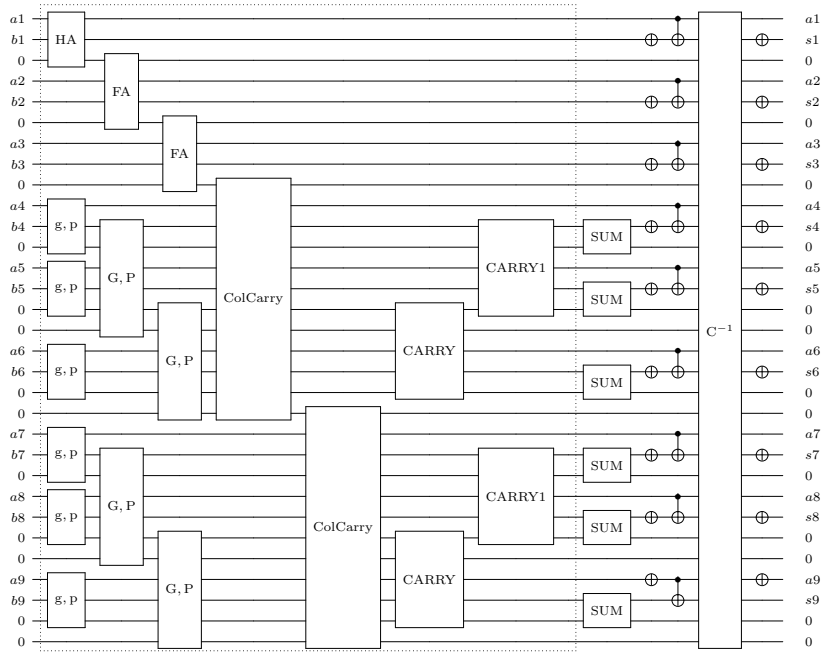
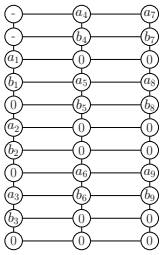


Fig. 2. The revised block diagram of a 2D 9-bit adder in [18] based on the blocks used in this paper. The critical path in this circuit is g,p-->G,P-->ColCarry-->ColCarry-->CARRY-->CARRY1-->SUM. The C^{-1} block is the reverse of the circuit shown in the dashed box. This reverse circuit with the NOTs and CNOTs shown are applied to clear ancillae in [18]. Except for ColCarry (Column_carry), the number of inputs and outputs for other modules are the same as the ones shown in this figure. In Column_carry, the number of inputs/outputs is 3 — i.e., the first line and the last two lines are actual inputs and outputs. Note that these three lines are neighbor in the 2D layout. The qubit placement for this 2D grid and their values during the computation (up to clearing ancillae) are given in Fig. 3.

Fig. 6(b). Accordingly, three SWAP gates in each G,P block can be saved. Fig. 7 shows the new circuits for Carry and Carry1. Note that some of G,P blocks are directly connected to the Carry (or Carry1) blocks without any interaction with Column_carry blocks. For such cases, we can apply the same mechanism.

- **G,P $\Rrightarrow$ G,P**: Each G,P block constructs two outputs based on (4) and (3) where $G[k\sqrt{n}+1, k\sqrt{n}+j]$ depends on $G[k\sqrt{n}+1, k\sqrt{n}+j-1]$ and $P[k\sqrt{n}+1, k\sqrt{n}+j]$ depends on $P[k\sqrt{n}+1, k\sqrt{n}+j-1]$. Since $G[k\sqrt{n}+1, k\sqrt{n}+j]$ is constructed first, we can use it to construct $G[k\sqrt{n}+1, k\sqrt{n}+j+1]$ in parallel to construction of $P[k\sqrt{n}+1, k\sqrt{n}+j-1]$. This can save one Toffoli and one SWAP. Fig. 8 shows the result of this optimization.



-	a_4	a_7
-	b_4, p_4, s_4	b_7, p_7, s_7
a_1	0, g_4, c_4	0, g_7, c_7
b_1, s_1	a_5	a_8
0, c_2	b_5, p_5, s_5	b_8, p_8, s_8
a_2	0, g_5, $P[4,5]$, c_4, c_5	0, g_8, $P[7,8]$, c_7, c_8
b_2, s_2	0, $G[4,5]$, $P[4,5]$	0, $G[7,8]$, $P[7,8]$
0, c_3	a_6	a_9
a_3	b_6, p_6, s_6	b_9, p_9, s_9
b_3, s_3	0, g_6, $P[4,6]$, c_4, c_6	0, g_9, $P[7,9]$, c_7, c_9
0, c_4	0, $G[4,6]$, c_7	0, $G[7,9]$, c_{10}

Fig. 3. The qubit placement for the 2D grid in Fig. 2 and their values during the computation.

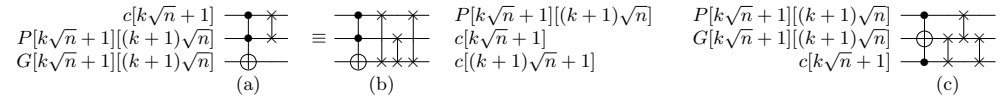


Fig. 4. (a) Circuit structure for Column_carry based on (5). Note that $c[(k-1)\sqrt{n}+1]$ and $P[(k-1)\sqrt{n}+1][k\sqrt{n}]$ are not adjacent (see Fig. 2). (b) Circuit in (a) with adjacent gates. (c) Circuit in (b) with relabelled qubits to show adjacent qubits.

4 Depth Analysis

In this section, we analyze the circuit depth of a 2D n-bit quantum adder based on the circuit structures proposed for each block.

- **Phase 1 — Half-adder+Full-adder**: We can execute Half-adder and the first two gates ($\mathcal{T}$+$\mathcal{C}$) in all Full-adders in parallel. This results in $1\mathcal{T}$+$1\mathcal{C}$+$(\sqrt{n}-1)(2\mathcal{S}$+$1\mathcal{C}$+$1\mathcal{T})$ time steps.
- **Phase 1 — g,p+G,P**: Each g,p block includes one Toffoli and one CNOT gates. Except for the first G,P block, the other $\sqrt{n}-2$ G,P blocks include 3 SWAPs and 1 Toffoli. The first G,P block includes two Toffoli and two SWAP gates. Altogether, circuit depth can be calculated as $(1\mathcal{T}$+$1\mathcal{C})$+$(2\mathcal{T}$+$2\mathcal{S})$+$(\sqrt{n}-2)(3\mathcal{S}$+$1\mathcal{T})$.
- **Phase 2 — Column_carry**: There are $\sqrt{n}-1$ Column_carry blocks in cascade. This results in $\sqrt{n}-1(1\mathcal{T}$+$3\mathcal{S})$ time steps.
- **Phase 3 — Carry + SUM**: There are $\sqrt{n}-2$ Carry blocks followed by one Carry1 block and one SUM block. Therefore, circuit depth is $(\sqrt{n}-2)(1\mathcal{T}$+$4\mathcal{S})$+$(3\mathcal{S}$+$1\mathcal{T})$+$1\mathcal{C}$.

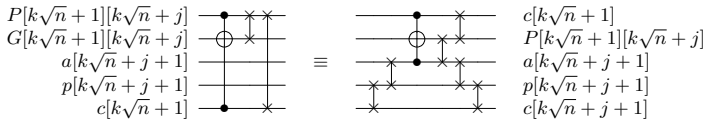


Fig. 5. Circuit structure for Carry based on (6). Inputs $a[k\sqrt{n}+j+1]$ and $p[k\sqrt{n}+j+1]$ are not used in the computation.

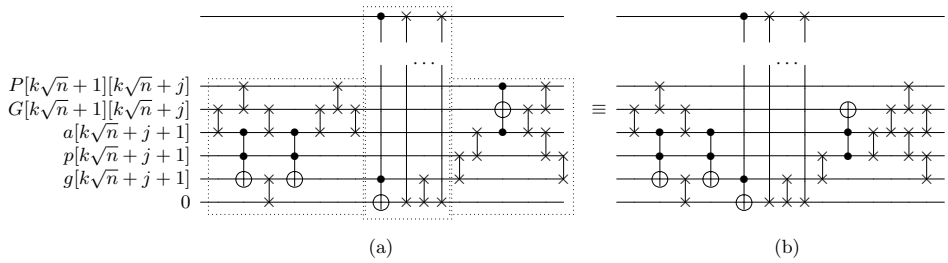


Fig. 6. (a) G,P, Column_carry, and Carry blocks in cascade. The three rightmost SWAP gates in G,P can be merged with gates in the Carry block to construct a new circuit shown in (b).

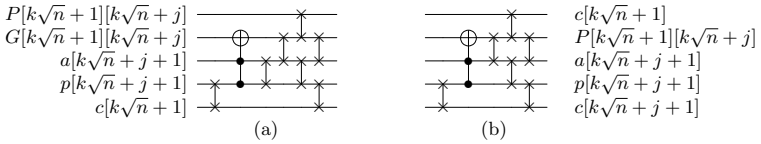


Fig. 7. New circuit structures for Carry (a) and Carry1 (b) based on the optimization shown in Fig 7. Note that the first SWAP gate can be executed in parallel with gates in the previous block (see Fig. 7).

Table 2 reports circuit depth for each component and the total depth in the proposed 2D quantum adder. As can be seen in this table, circuit depth is improved by a factor of $\frac{26}{35}$ (i.e., %24).

In [24], a new circuit for Peres with depth=5C+3 has been proposed (Fig. 10(a)). After inserting one CNOT (to have Toffoli) and two SWAP gates to have adjacent gates, one can use the new circuit with depth=6C+2$\mathcal{S}$+4 in order to further optimize the proposed 2D adder. Note that in [24], a circuit structure for Toffoli gate with depth=6C+2 has been proposed too, Fig. 9. However, working with Peres gate results in a more compact circuit in terms of the number of SWAP gates. Following this path results in depth=$92\sqrt{n}$+const for the proposed 2D quantum adder. Table 3 compares circuit depth based on different costs for Toffoli and SWAP gates.

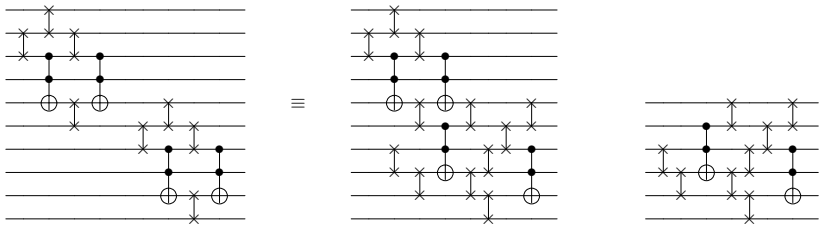


Fig. 8. Construction of $G[k\sqrt{n}+1, k\sqrt{n}+j+1]$ can be done in parallel to construction of $P[k\sqrt{n}+1, k\sqrt{n}+j-1]$ in two consecutive G,P blocks. The right circuit shows the new circuit structure for G,P (except for the first G,P block).

Table 2. Circuit depth for our blocks in 2D adder. Circuit depths for CNOT ($\mathcal{C}$), SWAP ($\mathcal{S}$), and Toffoli ($\mathcal{T}$) gates are considered as 1, 1, and 14 as done in [18].

Block	Circuit	Ours	[18]
Half-adder	$1\mathcal{T}+1\mathcal{C}$	15	15
Full-adder	$2\mathcal{S}+1\mathcal{C}+1\mathcal{T}$	17	32
g,p	$1\mathcal{T}+1\mathcal{C}$	15	15
G,P (first)	$2\mathcal{T}+2\mathcal{S}$	30	34
G,P (others)	$3\mathcal{S}+1\mathcal{T}$	17	34
Column_carry	$1\mathcal{T}+3\mathcal{S}$	17	18
Carry	$1\mathcal{T}+4\mathcal{S}$	18	18
Carry1	$3\mathcal{S}+1\mathcal{T}$	17	18
SUM	$1\mathcal{C}$	1	5
Phase1-1	$1\mathcal{T}+1\mathcal{C}+(\sqrt{n}-1)(2\mathcal{S}+1\mathcal{C}+1\mathcal{T})$	$17\sqrt{n}-2$	$32\sqrt{n}-17$
Phase1-2	$(1\mathcal{T}+1\mathcal{C})+(2(\mathcal{T}+2\mathcal{S})+(\sqrt{n}-2)(3\mathcal{S}+1\mathcal{T})$	$17\sqrt{n}+11$	$34\sqrt{n}-19$
Phase2	$(\sqrt{n}-1)(1\mathcal{T}+3\mathcal{S})$	$17\sqrt{n}-17$	$18\sqrt{n}-18$
Phase3	$(\sqrt{n}-2)(1\mathcal{T}+4\mathcal{S})+(3\mathcal{S}+1\mathcal{T})+1\mathcal{C}$	$18\sqrt{n}-18$	$18\sqrt{n}+1$
clearing ancillae	Phase1-2+Phase2+Phase3-SUM	$52\sqrt{n}-24$	$70\sqrt{n}-39$
2D Adder	Phase1-2+Phase2+Phase3+clearing ancillae+3	$104\sqrt{n}-46$	$140\sqrt{n}-72$

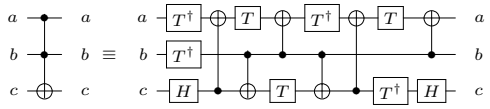


Fig. 9. Toffoli decomposition with depth 6C+2 [24].

5 Error Correction

To protect quantum information from errors due to e.g., noise or decoherence, quantum error correction (QEC) should be used in any large-scale quantum computation. In the recent years, various models for QEC have been proposed [20]. A common technique, known as concatenated quantum code, is to encode a logical qubit into the state of several physical qubits (e.g., 7 in Steane code and 9 in Bacon-Shor code [20], both for one level of concatenation).

Let assume each unitary operation should be followed by quantum error correction for proper computation. This results in an aggressive quantum error correction mechanism. In some circumstances, one may insert error correction after several operations, instead of each operation. Consider a quantum computation U with N_U logical operations which include only FT quantum gates. Moreover, assume that error correction for each FT gate requires N_E physical instructions. N_E includes SWAPs required for communication. Normally, N_E differs for various logical operations; however, we can consider the worst-case value among all FT gates. Working with concatenated quantum error correction

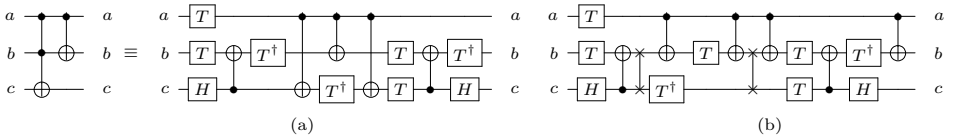


Fig. 10. (a) Peres decompositions with depth 5C+3 [24], (b) Toffoli with adjacent gates based on Peres decomposition (depth=6C+2$\mathcal{S}$+4).

Table 3. Circuit depth for the proposed adder and the one in [18] considering different costs for Toffoli and SWAP gates.

$\mathcal{T}$-depth=14,$\mathcal{S}$-depth=1		$\mathcal{T}$-depth=14,$\mathcal{S}$-depth=3		$\mathcal{T}$-depth=12,$\mathcal{S}$-depth=3		$\mathcal{T}$-depth=12,$\mathcal{S}$-depth=1	
Ours	[18]	Ours	[18]	Ours	[18]	Ours	[18]
$104\sqrt{n}$	$140\sqrt{n}$	$144\sqrt{n}$	$176\sqrt{n}$	$132\sqrt{n}$	$160\sqrt{n}$	$92\sqrt{n}$	$124\sqrt{n}$

techniques, the total physical gate count at concatenation level L can be estimated as $N_L = N_{L-1} + N_{L-1} \times N_E$ or $N_L \approx N_{L-1} \times N_E$. We have $N_0 = N_U$, and therefore, $N_L = N_U(N_E)^L$. Accordingly, besides the effect of the proposed approach on circuit depth, one can implement the proposed 2D adder with fewer physical gates — the reduction factor is 24/35.

6 Conclusion

In this paper, we considered a quantum adder on 2D quantum architectures. Our work in based on the results reported in [18] with several extensions. In particular, we optimized the building blocks of the 2D adder with focus on reducing the communication overhead required in 2D quantum architectures. Having optimized consecutive blocks, the proposed adder can execute expensive Toffoli gates concurrently in several locations. The suggested optimizations improve depth=$140\sqrt{n}+k_1$ in [18] to $92\sqrt{n}+k_2$ for constants k_1 and k_2. Considering the fact that the number of possible locations where an error can occur in a quantum commutating system depends on circuit depth $\times$ circuit width (i.e., number of qubits), the proposed adder can be affected by quantum error less, by a factor of 34%.

References

1. D. Cheung, D. Maslov, and S. Severini. Translation techniques between quantum circuit architectures. *Workshop on Quant. Inf. Proc.*, Dec 2007.
2. J. Díaz, J. Petit, and M. J. Serna. A survey of graph layout problems. *ACM Comput. Surv.*, 34(3):313–356, 2002.
3. B.-S. Choi and R. Van Meter. On the effect of quantum interaction distance on quantum addition circuits. *J. Emerg. Technol. Comput. Syst.*, 7(3):11:1–11:17, August 2011.
4. R. Beals et al. Efficient distributed quantum computing, arXiv:1207.2307v2, 2012.
5. Y. Takahashi, N. Kunihiro, and K. Ohta. The quantum Fourier transform on a linear nearest neighbor architecture. *Quant. Inf. Comput.*, 7:383–391, 2007.
6. D. Maslov. Linear depth stabilizer and quantum Fourier transformation circuits with no auxiliary qubits in finite neighbor quantum architectures. *Phys. Rev. A*, 76, 2007.
7. A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg. Implementation of Shor's algorithm on a linear nearest neighbour qubit array. *Quant. Inf. Comput.*, 4:237–245, 2004.
8. S. A. Kutin. Shor's algorithm on a nearest-neighbor machine. *Asian Conf. on Quant. Inf. Sci.*, 2007.

9. P. Pham and K. M. Svore. A 2D nearest-neighbor quantum architecture for factoring. *arXiv:1207.6655*, 2012.
10. A. G. Fowler, C. D. Hill, and L. C. L. Hollenberg. Quantum error correction on linear nearest neighbor qubit arrays. *Phys. Rev. A*, 69:042314.1–042314.4, 2004.
11. M. Arabzadeh, M. Saheb Zamani, M. Sedighi, and M. Saeedi. Depth-optimized reversible circuit synthesis. *Quant. Inf. Proc.*, 12(4):1677–1699, 2013.
12. M. Möttönen and J. J. Vartiainen. Decompositions of general quantum gates. *Ch. 7 in Trends in Quantum Computing Research, NOVA Publishers, New York*, 2006.
13. V. V. Shende, S. S. Bullock, and I. L. Markov. Synthesis of quantum-logic circuits. *IEEE Trans. CAD*, 25(6):1000–1010, June 2006.
14. M. Saeedi, M. Arabzadeh, M. Saheb Zamani, and M. Sedighi. Block-based quantum-logic synthesis. *Quant. Inf. Comput.*, 11(3-4):0262–0277, 2011.
15. M. Saeedi, M. Saheb Zamani, M. Sedighi, and Z. Sasanian. Reversible circuit synthesis using a cycle-based approach. *J. Emerg. Technol. Comput. Sys.*, 6(4):13:1–13:26, 2010.
16. M. Saeedi, R. Wille, and R. Drechsler. Synthesis of quantum circuits for linear nearest neighbor architectures. *Quant. Inf. Proc.*, 10(3):355–377, 2011.
17. Y. Hirata, M. Nakanishi, S. Yamashita, and Y. Nakashima. An efficient conversion of quantum circuits to a linear nearest neighbor architecture. *Quant. Inf. Comput.*, 11(1–2):0142–0166, 2011.
18. B.-S. Choi and R. Van Meter. A $\sqrt{n}$-depth quantum adder on the 2D NTC quantum computer architecture. *J. Emerg. Technol. Comput. Syst.*, 8(3):24:1–24:22, August 2012.
19. T. Szkopek et al. Threshold error penalty for fault-tolerant quantum computation with nearest neighbor communication. *Nanotechnology, IEEE Transactions on*, 5(1):42 – 49, jan. 2006.
20. M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.
21. I. L. Markov and M. Saeedi. Constant-optimized quantum circuits for modular multiplication and exponentiation. *Quantum Info. Comput.*, 12(5-6):361–394, May 2012.
22. I. L. Markov and M. Saeedi. Faster quantum number factoring via circuit synthesis. *Phys. Rev. A*, 87:012310, Jan 2013.
23. V. V. Shende and I. L. Markov. On the CNOT-cost of TOFFOLI gates. *Quant. Inf. Comput.*, 9(5-6):461–486, May 2009.
24. M. Amy, D. Maslov, M. Mosca, and M. Rötteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans. CAD, arXiv:1206.0758v3*, 2013.

Acknowledgements

Authors were supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20165. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

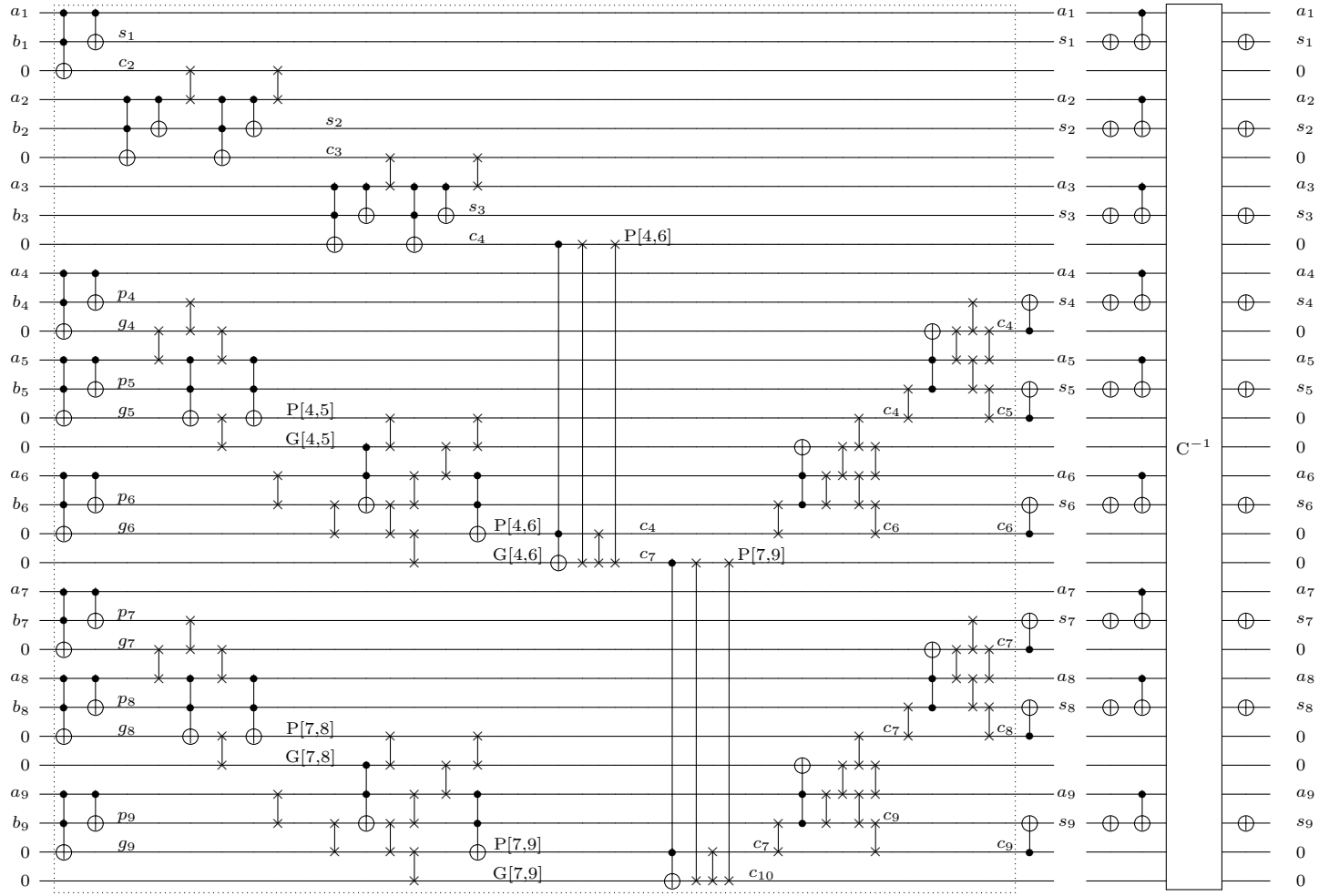


Fig. 11. A 9-bit adder based on the proposed blocks. Carry, $G_{i,j}$, p_i, and g_i values are shown in this figure. The C^{-1} block is the reverse of the circuit shown in the dashed box applied with the NOTs and CNOTs shown to clear ancillae. All gates use adjacent gates in the 2D layout. For qubit locations see the table in Fig. 3.