Space-depth tradeoffs in parity synthesis for quantum computing

Arya Maheshwari

Princeton University

Advisor: Dr. Ali Javadi-Abhari (IBM Quantum)
PACM Reader: Prof. Noga Alon (Princeton University)

Introduction: Quantum compilation

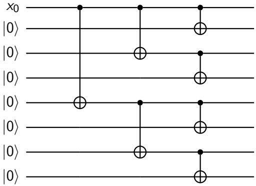
Current quantum computing (QC): large gap in theory vs. practice

1. Noisy operations (gates) on qubits ⇒ limited number of operations
2. Qubit *decoherence* ⇒ limited execution time

Crucial for near-term QC: optimize circuit resources required to implement a quantum algorithm = **quantum synthesis/compilation**

Various metrics of interest:

- **# qubits used** ("space")
- gate count (re: (1))
- **circuit depth**: number of *layers* of gates (re: (2))



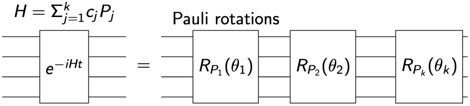
Gate count: 7. Circuit Depth: 3. Number of qubits: 8 (1 *data* qubit, 7 *ancilla* qubits)

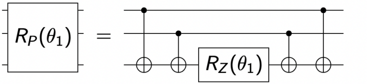
Basic compilation ideas: reorder commuting blocks, gate cancellation

Motivation: Hamiltonian simulation → parity synthesis

Problem: Hamiltonian simulation → implementing Pauli rotation sequence

- Each Pauli rotation ⇒ compute specific *parity* via CNOTs + single-qubit rotation





Compilation bottleneck: CNOT gates

- ⇒ Focus optimization on CNOT depth, i.e. on just parity computation component of implementing Pauli rotations

Narrow problem: how to synthesize a *parity network* in minimal depth?

- Parity network = CNOT circuit where all required parities appear *somewhere* in the circuit. (e.g. [VMGDB22])

Problem Statement: ParallelParities

This project: a variant of parity network synthesis:
Require that all parities be present *simultaneously* in the circuit.

- Model: Assume access to *ancilla* qubits; want to store parities onto these ancilla
- Motivations: new approaches to Hamiltonian simulation, "CNOT+T" circuits.

Problem: ParallelParities

Input: n data qubits $\mathbf{x} = \{x_1, ..., x_n\}$; p parities $\{f_1(\mathbf{x}), ..., f_p(\mathbf{x})\}$ over the data qubits; m ancilla

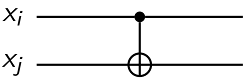
Output: In minimal depth, synthesize the p parities onto ancilla so that they are present *simultaneously*

Hope: leverage ancilla (more "space") for parallelization: larger $m \Rightarrow$ smaller depth required to implement ParallelParities?

Our goal: Understand *space-depth tradeoff* in ParallelParities better.

Preliminaries and Model

- *Data qubits*: $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, $x_i \in \mathbb{F}_2$
- *Parity* $f_j(\mathbf{x})$: sum of some x_i's (over $\mathbb{F}_2$), e.g. $f_j(\mathbf{x}) = x_1 \oplus x_2 \oplus x_4$.
- *Ancilla qubit*: "helper" qubit, starts in zero state $|0\rangle$.
- CNOT (controlled-NOT): 2-qubit gate:



$$\text{CNOT}(x_i, x_j) = (x_i,\ x_i \oplus x_j)$$

- **Key abstraction:** CNOT circuits over n qubits $\leftrightarrow$ linear reversible transformations over $\mathbb{F}_2$ (i.e. $GL_n(\mathbb{F}_2)$).
  - Reason: $\text{CNOT}(x_i, x_j)$ action given by $R_{ij}\mathbf{x}$ for row addition matrix R_{ij}.
- $\Rightarrow$ constructing CNOT circuit on n qubits = "n-qubit linear reversible circuit synthesis," viewed as algorithmic task on Boolean matrices.

Roadmap

1. Existing approach 1: minimizing ancilla (space)
2. Existing approach 2: minimizing depth
3. Our approach: a controllable space-depth tradeoff

Existing idea: minimizing ancilla

Minimal use of space: $m = p$ ancilla (one for each parity)

Approach ([BBV$^+$21]): *isometry synthesis* via Block algorithm.

- Represent *parity state* as $(n+p) \times n$ matrix.
- Goal: CNOT circuit C (repr. by matrix $\in \mathbb{F}_2^{(n+p)\times(n+p)}$) to transform $A_{in} \to A_{out}$.
- Use n-qubit linear reversible synthesis routine as a *blackbox input* (with some depth upper bound $d(n)$).

$$A_{in} = \overbrace{\left[\begin{array}{c} \mathbb{I}_n \\ \hdashline \mathbf{0} \end{array}\right]}^{n} \begin{array}{l} \}\, n \\ \}\, p \end{array} \qquad A_{out} = \overbrace{\left[\begin{array}{c} \mathbb{I}_n \\ \hdashline f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_i(\mathbf{x}) \\ \vdots \\ f_p(\mathbf{x}) \end{array}\right]}^{n} \begin{array}{l} \}\, n \\ \}\, p \end{array}$$

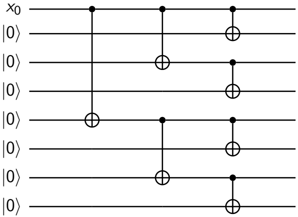
Lemma 1 (Block algorithm [BBV$^+$21])

A CNOT circuit $C \in \mathbb{F}_2^{(n+p)\times(n+p)}$ such that $A_{out} = CA_{in}$ can be synthesized in depth upper bounded by $d(n) + 2\lceil \log(\frac{p}{n} + 1) \rceil$.

Existing idea: minimizing depth

Key tool: logarithmic depth spreading:

- Task: spreading some x_i to some set of k $|0\rangle$ wires.
- Naive approach: k sequential CNOTs. Depth: k.
- Better: tree-like spreading. Depth: $\lceil \log{(k+1)} \rceil$.



Spreads a qubit x_i to $k = 7$ wires in depth $\lceil \log{(k+1)} \rceil = 3$.

Lemma 2

Given access to $m = p \cdot n$ ancilla, the p parities over n data qubits in ParallelParities *can be synthesized in $\lceil \log(p+1) \rceil + \lceil \log n \rceil$ depth.*

Proof: Create p registers of n ancilla each. Then, simple two-step procedure:

1. Spread each x_i to each register, *in parallel*: $\log(p+1)$ depth.
2. Compute jth parity over n ancilla in jth register, *in parallel*: $\log(n)$ depth.

Our approach: controllable space-depth tradeoff

So far: two ends of spectrum:

1. Lemma 1: only p ancilla; depth $d_a(n,p) := d(n) + 2\lceil \log(\frac{p}{n}+1)\rceil$, where $d(n) =$ upper bound on n-qubit linear reversible circuit depth.
2. Lemma 2 np ancilla; reduces depth to $d_b(n,p) := \lceil \log(p+1)\rceil + \lceil \log(n)\rceil$.

Our contribution: framework to control space-depth tradeoff via a parameter c.

Theorem 3 (Main result: c-controlled synthesis)

The p parities defined over n qubits in ParallelParities *can be synthesized using $m = c \cdot p$ ancilla in depth at most*

$$d(n,p;c) = d\left(\frac{n}{c}\right) + 2\left\lceil \log\left(\frac{cp}{n}+1\right)\right\rceil + \lceil \log(c)\rceil$$

for any divisor c of n.

Base cases: $c = 1, c = n$: recovers Lemma 1 and 2 (up to small constant).

Key idea: split each parity into c pieces and run c Block instances in parallel.

Summary

- Introduced the ParallelParities problem (a subroutine motivated by e.g. Hamiltonian simulation)
    - Compilation goal: minimize depth vs. ancilla use
- Existing optimization ideas: two extremes, no clear way to interpolate
- Our approach: simple framework for *controlling* space-depth tradeoff via a tunable parameter c.
- Utility: Enables *instance-specific* allocation of space and depth costs: practitioners can choose how much of each cost to tolerate.

Acknowledgements

- Dr. Ali Javadi-Abhari, Dr. Simon Martiel, and IBM Quantum
- Prof. Noga Alon
- Prof. Paul Seymour, Ms. Victoria Beltra, and PACM program

Thank you! Questions?

Timothée Goubault de Brugière, Marc Baboulin, Benoît Valiron, Simon Martiel, and Cyril Allouche.
Reducing the depth of linear reversible quantum circuits.
IEEE Transactions on Quantum Engineering, 2:1–22, 2021.

Vivien Vandaele, Simon Martiel, and Timothée Goubault De Brugière.
Phase polynomials synthesis algorithms for nisq architectures and beyond.
Quantum Science and Technology, 7(4):045027, Oct 2022.