東南大學 SOUTHEAST UNIVERSITY

Towards Fault-tolerant Design of Quaternary Quantum Arithmetic

Yuchen Zhu, Ruixuan Yang, Yuhang Gu, Fangtian Gu, Lingyi Kong, and He Li*

helix@seu.edu.cn

Heterogeneous Intelligent and Quantum Computing (HIQC) Lab
School of Electronic Science & Engineering
Southeast University, Nanjing, China

20 Aug. 2024

Outline

Background

Quaternary Mapping Method

Q^2FA Design Strategy

Evaluation of Q^2FAs

Outline

Background

Fast-growing of Quantum Computers

- Integer factorization and database search ✓
- Variational Quantum Eigensolver (VQE) for quantum chemistry ✓
- Hybrid quantum-classical optimization algorithms ✓

Limitations on Fixed Computation Radix

- Unable to discriminate multiphase shift ✗
- Limited space for Quantum Error Correction (QEC) ✗
- Fail to support problem-tailored encodings ✗

Our Work

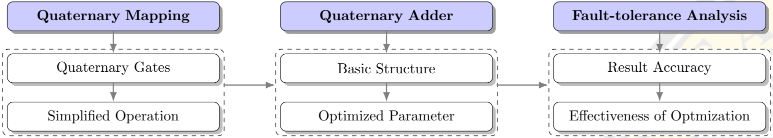

Fig 1: Overview of Our Work

Our Work

- A mapping method to realize arbitrary quaternary gates on binary circuits and simplify certain operations
- Novel designs of quaternary quantum full adders (Q^2FAs), from basic structure to circuits with optimized metrics
- Fault-tolerance analysis of proposed Q^2FA designs and highlight the effectiveness of our design

Outline

Quaternary Mapping

- Quaternary operations based on Galois Field (GF) to ensure reversibility
- Examples of quaternary operations: $Z(03) = 2A^2 + 3$, $Z(0321) = 3A^2 + 3$, $Z(23) = A^2$
- Quaternary Mapping: $|0\rangle \to |00\rangle$, $|1\rangle \to |01\rangle$, $|2\rangle \to |10\rangle$, $|3\rangle \to |11\rangle$

Tab 1: Truth Table of GF(4) Arithmetic.

$\oplus$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$	$\odot$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$
$\|0\rangle$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|3\rangle$	$\|2\rangle$	$\|1\rangle$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$
$\|2\rangle$	$\|3\rangle$	$\|0\rangle$	$\|1\rangle$	$\|2\rangle$	$\|2\rangle$	$\|3\rangle$	$\|1\rangle$
$\|3\rangle$	$\|2\rangle$	$\|1\rangle$	$\|0\rangle$	$\|3\rangle$	$\|3\rangle$	$\|1\rangle$	$\|2\rangle$

Quaternary Mapping

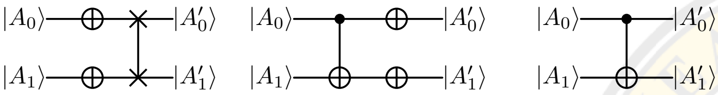

Fig 2: Z(03), Z(0321), and Z(23) Gates after Mapping

Feasible for large-scale quantum computers with simplified operations

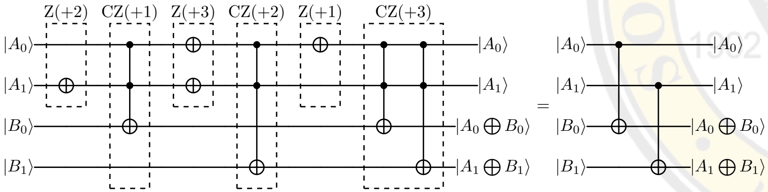

Fig 3: Common approach to quaternary CX and simplified operation

Outline

From QFA to Q^{2}FA

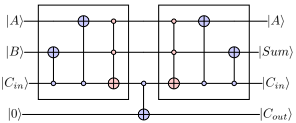

Fig 4: QFA

- $Sum = A \bigoplus B \bigoplus C_{in}$
- $Carry = AB + BC_{in} + AC_{in}$

1. Repetitive X gates to ensure reversibility
2. Modifications in the 2nd half adder

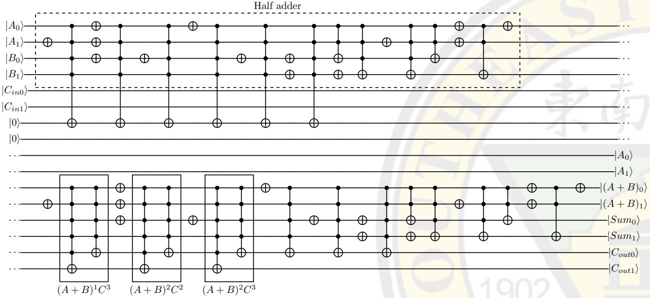

Fig 5: Q^{2}FA-1

- $Sum = (A + B) \mod 4$
- $Carry = A^1B^3 + A^2B^2 + A^2B^3 + A^3B^1 + A^3B^2 + A^3B^3$

Consume 8 qubits, 6 2CX gates, 4 3CX gates, 12 4CX gates, and 3 5CX gates.

Logic Optimization

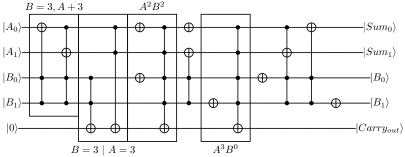

Fig 6: Logical optimized half adder

When either of the inputs is in $|3\rangle$, the carry is most likely to be $|1\rangle$. Additional circuits are applied for the rest of the cases.

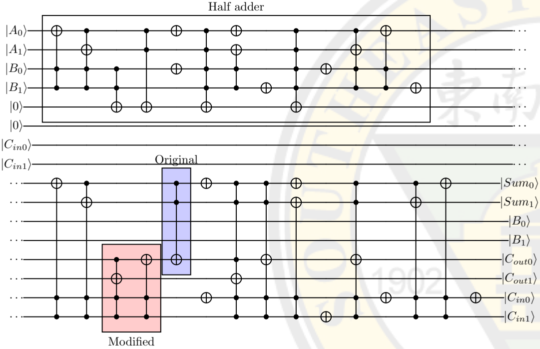

Fig 7: Q^2FA-2

Consume 8 qubits, 10 2CX gates, 4 3CX gates, 4 4CX gates, and 2 5CX gates.

Superposition Inspired Design

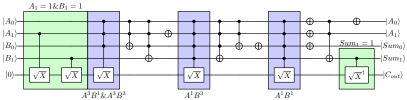

Fig 8: Logical optimized half adder

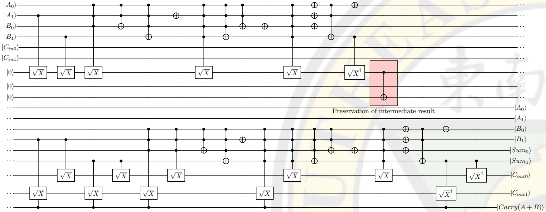

Fig 9: Q^2FA-3

Empirically, the green parts satisfy most of the cases. The rest of the cases are handled by the purple parts.

For Q^2FA-3, an additional qubit is required to store the intermediate carry result.

Consume 9 qubits, 9 2CZX gates, 6 3CZX gates, 5 4CZX gates, and 1 5CZX gates.

Outline

Background

Quaternary Mapping Method

Q^2FA Design Strategy

Evaluation of Q^2FAs

Evaluation Method

Metrics

1. Qubit consumption
   Most important metric for NISQ era
2. T-count
   Nonclifford gates are the most expensive gates which can be represented by T-count and is closely related with multi-controlled gates
3. T-depth
   Number of layers of T gates, which is crucial because of the limited relaxation time (T_1) and dephasing time (T_2)

Method

Classical emulation + Hardware implementation

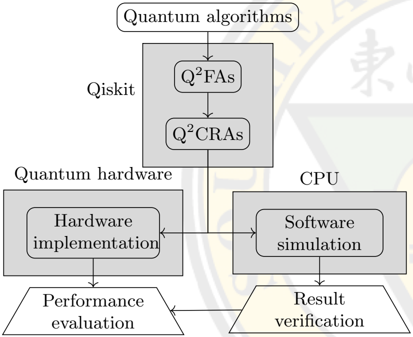

Fig 10: Evaluation method

Evaluation result

Tab 2: Comparison of various Q^2FAs.

Type	Qubits (Garbage)	X	CX	2CX	3CX	4CX	5CX	T-count
		Gates						
Q^2FA-1	8(4)	28	0	6	4	12	3	483
Q^2FA-2	8(4)	12	0	10	4	4	2	292
Q^2FA-3	9(5)	12	7	9	6	5	1	303
		Circuit depth						
		X	CX	2CX	3CX	4CX	5CX	T-depth
		19	0	6	4	12	3	93
		8	0	10	4	4	2	68
		8	7	9	5	5	1	67

Analysis

- Q^2FAs follow the same design strategy, i.e., from half adder to full adder
- Q^2FA-2 generally utilizes fewer gates than the other two, owing to the optimization of the carry propagation path
- Q^2FA-3 has the lowest T-depth, which is attributed to replacing multi-controlled X gates with multi-controlled ZX gates.

Evaluation result

Tab 3: Comparison of various Q^2CRAs.

QFA Type	Qubits (Garbage)				Gates			
		X	CX	2CX	3CX	4CX	5CX	T-count
Q^2FA-1	$6p$+2($4p$)	$28p$	0	$6p$	$4p$	$12p$	$3p$	$483p$
Q^2FA-2	$6p$+2($4p$)	$12p$	0	$10p$	$4p$	$4p$	$2p$	$292p$
Q^2FA-3	$7p$+2($5p$)	$12p$	$7p$	$9p$	$6p$	$5p$	$1p$	$303p$
					Circuit depth			
		X	CX	2CX	3CX	4CX	5CX	T-depth
		$19p$	0	$6p$	$4p$	$12p$	$3p$	$93p$
		$8p$	0	$10p$	$4p$	$4p$	$2p$	$68p$
		$8p$	$7p$	$9p$	$5p$	$5p$	$1p$	$67p$

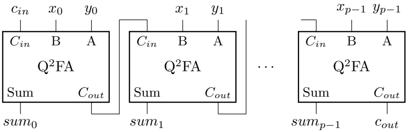

Fig 11: Schematic of quanternary quantum carry-ripple adder (Q^2CRA)

Carry-ripple adders, as one of the most classical least-significant digit-first (LSDF) arithmetic operators, are composed of an array of FAs. Implemented with Q^2FA-2, Q^2CRAs exhibit a significant advantages of circuit depth compared to the design with Q^2FA-1.

SOUTHEAST UNIVERSITY

Fault-tolerant Evaluation

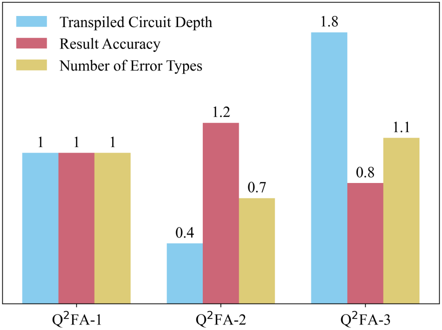

Fig 12: Fault-tolerant performance of Q^2FAs

- Setup: All results are obtained under the configuration of $ibm\_brisbane$
- Performance: Q^2FA-2 exhibits a 1.2× higher result accuracy and a 1.5× lower number of error types
- Analysis: Q^2FA-3 shows a worse fidelity in contrary to the metrics, which is attributed to the basis gates and the mismatch between T_1 and T_2

Effectiveness of Optimization

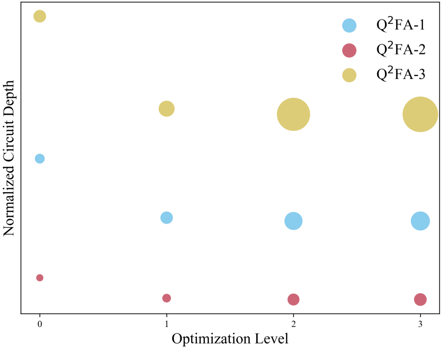

Fig 13: Relationship between optimization level of the compiler and the transpiled circuit depth.

- Setup: Qiskit compiler is adopted to optimize the circuit under various optimization levels
- Performance: the circuit depth decreases with the optimization level, along with an enormous increase in compilation time (denoted as the area of each point)

Thanks for Listening.