

A 22-nm Surface Code Decoder Using Greedy Algorithm

Junichiro Kadomoto
The University of Tokyo
Tokyo, Japan
kadomoto@mtl.t.u-tokyo.ac.jp

Ren Aoyama
Kyoto Institute of Technology
Kyoto, Japan
raoyama@vlsi.es.kit.ac.jp

Kazutoshi Kobayashi
Kyoto Institute of Technology
Kyoto, Japan
kazutoshi.kobayashi@vlsi.es.kit.ac.jp

Abstract—To realize a fault-tolerant quantum computer, a quantum error decoder that can handle a large number of qubits with high speed is required. This paper demonstrates an ASIC implementation of a quantum error decoder based on a greedy algorithm. The simple algorithm and microarchitecture enable a low-power and compact design. A test chip is fabricated using 22-nm CMOS technology, and its operation is verified through empirical evaluation.

Index Terms—quantum computing, quantum error correction, ASIC

I. INTRODUCTION

Fault-tolerant quantum computers with many qubits have the potential to solve numerous problems that are currently intractable for classical computers in a practical timeframe [1]. Achieving practical applications requires a large number of error-resistant qubits. This is accomplished by forming logical qubits, which have lower error rates, from multiple physical qubits, with many such logical qubits enabling fault-tolerant quantum computation [2]. The method of using multiple physical qubits to construct logical qubits and reduce errors through redundancy is known as quantum error correction codes.

Among the various quantum error correction codes, the surface code is considered one of the most promising [3]. The surface code can be implemented with a simple two-dimensional lattice of physical qubits, where only adjacent qubits are coupled, enabling efficient quantum computation. Efficient decoding of the surface code is known to be formulated as a minimum-weight perfect matching problem on a graph. Consequently, implementing a fault-tolerant quantum computer requires a classical decoder capable of solving such problems.

Fig. 1 illustrates an example of the architecture of a fault-tolerant quantum computer, including the decoder. In this paper, the error correction component within this architecture is referred to as the *Error decoder*, or simply “decoder.” A fault-tolerant quantum computer comprises a *Front-end* and a *Back-end*. The *Front-end* is responsible for directly controlling the qubits, while the *Back-end* performs error correction and other computation-related tasks. The *Instruction memory* stores the instructions to be executed, while the *Instruction decoder* decodes these instructions and forwards them to subsequent units. The *Mapper* assigns operations to physical qubits, and

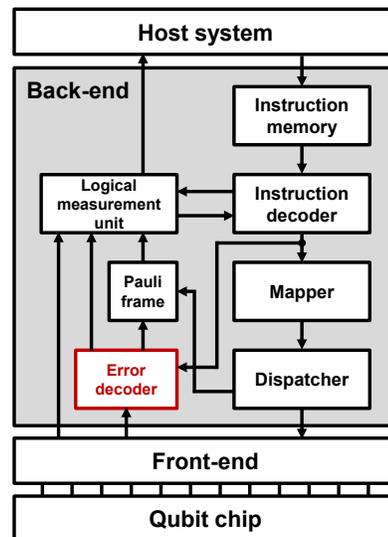


Fig. 1. Overall architecture of a fault-tolerant quantum computer.

the *Dispatcher* sends this information to the *Front-end* at appropriate times. The *Error decoder*, the focus of this paper, decodes measurement results, known as syndromes, from the physical qubits. Based on this decoded information, logical qubit measurements are performed within the *Pauli frame* and the *Logical measurement unit*.

To realize practical fault-tolerant quantum computers, it is essential to develop fast decoding methods for quantum error correction codes and to handle a large number of qubits [4]. Qubits are highly susceptible to decoherence due to interactions with their environment, making it crucial for decoders to rapidly detect and correct errors. Furthermore, such operations must be scalable to accommodate a large number of qubits for practical applications.

As a result, there has been ongoing research and development of fast and scalable decoders, including custom hardware implementations using FPGAs [5]. Among these custom hardware solutions, implementing application-specific integrated circuits (ASICs) through a mass-producible CMOS manufacturing process, along with optimization strategies, has been identified as a promising approach. Simulation-based evaluations and layout examples have demonstrated the

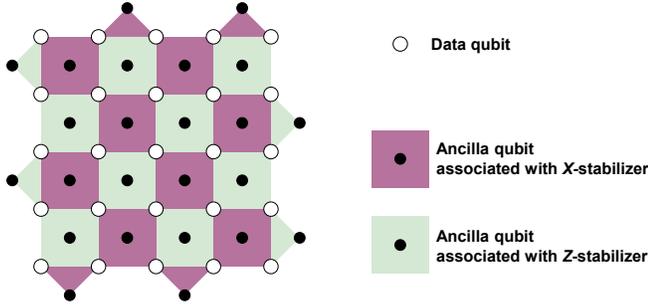


Fig. 2. Distance 5 rotated surface code.

potential of this approach [6]. However, there are only a few reports of decoders using ASICs, and to our knowledge, there has been no reported evaluation of a manufactured ASIC for quantum error decoding.

This paper presents measurement results of a surface code decoder test chip fabricated using 22-nm CMOS technology. The decoder employs the fast greedy algorithm as its decoding method and features a centralized memory architecture that enables implementation with low power and small area requirements. Although the presented results do not involve integration with actual qubits, this is, to our knowledge, the first reported evaluation of a manufactured ASIC implementation of a quantum error decoder.

II. BACKGROUND

A. Surface Code

The surface code is one of the most prominent quantum error correction codes designed for two-dimensional nearest-neighbor architectures [3]. Its physical implementation requires only a two-dimensional lattice arrangement of physical qubits, along with couplings between adjacent qubits. It can tolerate a physical qubit error rate of approximately 1% [7], which is several orders of magnitude higher than other promising codes that are also compatible with two-dimensional lattice qubit arrangements [8]. Additionally, quantum computation methods feasible on the surface code have been proposed [9], [10]. Given its simplicity and the ability to operate with physical qubits exhibiting relatively high error rates, the surface code is considered a promising candidate for practical quantum computation.

An example of a surface code with a code distance of $d = 5$ is shown in Fig. 2. Here, a variant of the surface code known as the rotated surface code is illustrated. The surface code consists of two types of qubits: data qubits, which are used for computation, and ancilla qubits, which are used to detect errors in the data qubits. Data qubits are placed at the vertices of the lattice, while ancilla qubits are positioned on the lattice faces. By measuring the ancilla qubits, errors in the surrounding data qubits can be identified. Ancilla qubits associated with Z -stabilizers detect bit-flip (X) errors, while those associated with X -stabilizers detect phase-flip (Z) errors. Notably, ancilla

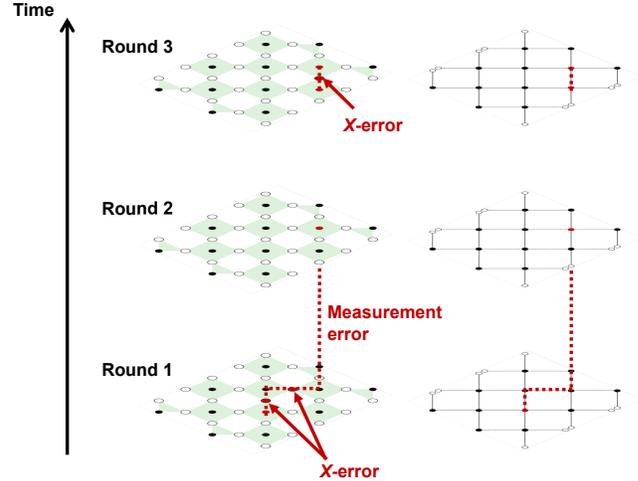


Fig. 3. Decoding graph.

qubits will not detect errors if an even number of surrounding data qubits experience errors.

B. Decoding Algorithm

In surface codes, the problem of estimating the error location from the syndrome can be formulated as a minimum-weight perfect matching (MWPM) problem, under the condition of minimizing the number of physical qubits experiencing errors. This approach is known to achieve high accuracy with low computational complexity [11]. To perform decoding, a decoding graph is constructed from the syndrome, as illustrated in Fig. 3. In this graph, the vertices represent the measurement outcomes of ancilla qubits, while the edges correspond to data qubits. By connecting pairs of error-detected ancilla qubits via the shortest paths on this graph and ensuring all vertices are paired, the location of the physical qubits experiencing errors can be inferred with minimal error count. This process is equivalent to solving an MWPM on a graph where each vertex represents an error-detecting ancilla qubit, and the edge weights correspond to the Manhattan distances between them.

In addition to errors on each data qubit, measurement errors can also occur on ancilla qubits. By connecting the decoding graphs from each measurement round to form a three-dimensional decoding graph, it becomes possible to estimate error locations while considering these measurement errors. For a code distance d , the minimum number of measurement rounds required is d . Fig. 3 only considers ancilla qubits associated with Z -stabilizers; however, the same approach applies to ancilla qubits associated with X -stabilizers. By solving these problems concurrently, it is also possible to handle Z -errors.

The Blossom algorithm has been proposed as an exact solution for MWPM, with a time complexity of $\mathcal{O}(|V|^2|E|)$ for a

given graph $G = (V, E)$. In addition, a faster approximation of MWPM using the greedy algorithm has been suggested [12], with a time complexity of $\mathcal{O}(|V|^2)$. Another example is a fast algorithm based on the union-find data structure [13], which can also be interpreted as an MWPM approximation algorithm [14].

III. RELATED WORK

Microarchitectures [15], [16] and hardware implementations [5], [17] for decoders targeting surface codes have been proposed in previous studies.

QECool [15] is a decoder based on a greedy algorithm, proposing a microarchitecture designed for implementation using SFQ logic. Its performance is evaluated through simulation studies. Additionally, [16] presents a decoder microarchitecture that utilizes an algorithm based on the union-find data structure.

WIT-Greedy [17] is an FPGA-based decoder that utilizes a greedy algorithm. It incorporates a design that accounts for variations in error rates, achieving a latency of 33.6 ns per measurement for a code distance of $d = 11$. Meanwhile, Helios [5] is an FPGA-based decoder leveraging the union-find data structure, with empirical evaluations demonstrating its performance. Through parallelization, it achieves a low latency of 19.3 ns per measurement for a code distance of $d = 51$.

While these prior studies have achieved high-speed decoding, they come with significant hardware resource requirements. For instance, [17] reports that memory capacity scales with the code distance as $\mathcal{O}(d^6)$, while [5] shows that hardware area scales as $\mathcal{O}(d^3)$ with the code distance. In contrast, this paper adopts a simpler greedy algorithm along with a centralized memory structure, aiming to achieve a balance between high-speed operation, low power consumption, and small area.

IV. SURFACE CODE DECODER USING GREEDY ALGORITHM

A. Algorithm

The greedy algorithm employed in the proposed decoder is shown in Algorithm 1. After converting the syndrome into a decoding graph, the constructed graph G is processed by iteratively selecting the pair of nodes with the minimum weight between them. This process continues until all nodes are paired.

B. Microarchitecture

Fig. 4 illustrates the block diagram of the proposed decoder. To achieve high-speed operation and a compact design in ASICs, a simple architecture with a centralized memory structure is employed. The syndrome is initially transformed into a decoding graph, which is then stored in a queue-structured memory. Each memory entry contains the coordinates of each node in the three-dimensional lattice decoding graph. Two entries corresponding to two nodes are sequentially read from memory, and the Manhattan distance between them is

Algorithm 1 Greedy algorithm

Require: Graph $G = (V, E)$ with edge weights $w(e)$ for $e \in E$

Ensure: A set of edges that forms a matching M

- 1: Initialize M as an empty set
 - 2: Mark all vertices in V as unmatched
 - 3: **while** there exists an unmatched vertex in V **do**
 - 4: Set $e_{\min} = \infty$ and $u_{\min} = v_{\min} = \text{null}$
 - 5: **for** each vertex $u \in V$ **do**
 - 6: **for** each vertex $v \in V$ such that $u \neq v$ and both u and v are unmatched **do**
 - 7: Find edge $e = (u, v)$ with weight $w(e)$
 - 8: **if** $w(e) < w(e_{\min})$ **then**
 - 9: Update $e_{\min} = e$ and $(u_{\min}, v_{\min}) = (u, v)$
 - 10: **end if**
 - 11: **end for**
 - 12: **end for**
 - 13: **if** $e_{\min} \neq \infty$ **then**
 - 14: Add edge e_{\min} to M
 - 15: Mark vertices u_{\min} and v_{\min} as matched
 - 16: **end if**
 - 17: **end while**
 - 18: **return** M
-

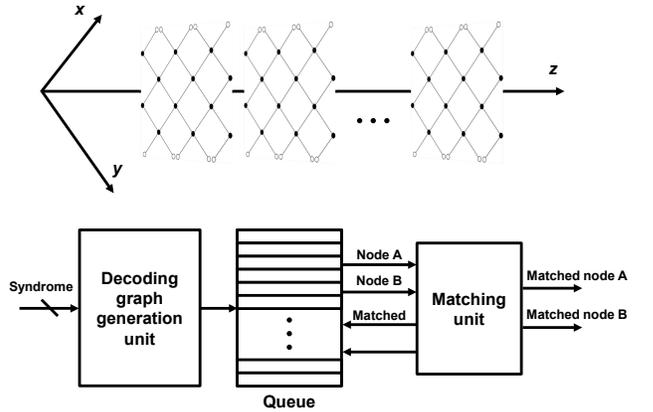


Fig. 4. Microarchitecture of the proposed decoder.

compared iteratively. Nodes that have already been matched are skipped, and this process continues until all nodes are paired.

V. MEASUREMENT RESULTS

A. Overview of the Test Chip

Fig. 5 shows the micrograph of the test chip, which was fabricated using 22-nm CMOS technology. The test chip consists of the designed decoder and a general-purpose control processor. The chip measures $1.5 \text{ mm} \times 2.0 \text{ mm}$, with the decoder occupying an area of 0.0235 mm^2 . The control processor is a 32-bit general-purpose processor based on the RISC-V ISA. The decoder receives syndrome-mimicking data from the control processor, performs the decoding, and then

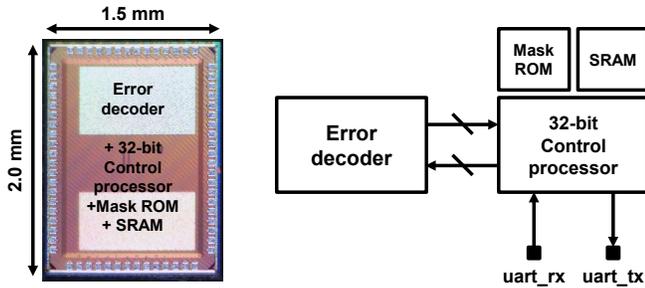


Fig. 5. Chip micrograph and the block diagram.

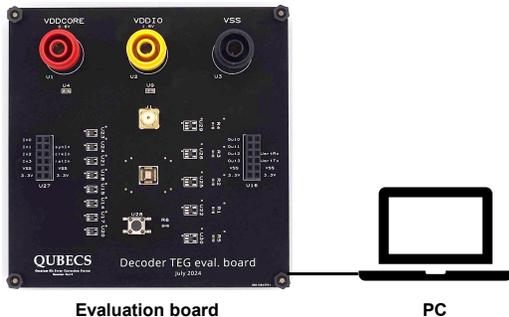


Fig. 6. Evaluation setup.

sends the results back to the control processor, where they can be verified.

Fig. 6 shows the evaluation board and measurement setup, with the test chip mounted on the board. Software was loaded onto the control processor via UART from a PC, and the operation of the decoder was verified through the control processor. Due to constraints in the measurement setup, the decoder was evaluated at an operating frequency of 10 MHz.

B. Decoding Performance

Syndrome-mimicking data was sent from the control processor to the decoder, and the results were verified to match those from simulations. The decoder operated at 10 MHz, with a measured power consumption of 0.16 mW. Based on the placement and routing results, the maximum operating frequency of the decoder is estimated to be 100 MHz, at which the power consumption is projected to be 1.6 mW. The decoder requires 1275 cycles for a single decoding operation, resulting in an estimated latency of 12.75 μ s.

To the best of our knowledge, the only prior work on ASIC-implemented decoders is presented in [6]. In that study, a decoding algorithm based on the union-find data structure was employed, with evaluations of power consumption and performance conducted through simulations. The study assumed a 12-nm FinFET technology, achieving a power consumption of 7.85 mW, a latency of 0.24 μ s per d-round measurement, and an area of 0.064 mm² under a code distance of $d = 23$. In contrast, our work adopts a simpler greedy algorithm and evaluates power consumption and performance using an

actual fabricated ASIC. For a 22-nm CMOS technology, the estimated power consumption was 1.6 mW, with an estimated latency of 12.75 μ s per d-round measurement and an area of 0.0235 mm² under a code distance of $d = 15$.

While it is important to note the differences in code distance and the impact of the chosen algorithm on decoding accuracy, our decoder achieves lower power consumption and area. However, it falls short in latency compared to prior work. This could potentially be improved by employing more advanced process nodes or enhancing microarchitecture to increase parallelism. The accuracy and latency requirements for decoders vary based on the characteristics of the qubits in use. Future research should therefore focus on discussions of suitable algorithms and microarchitectures, based on evaluation results involving actual qubits.

VI. CONCLUSION

This paper presents an ASIC implementation of a quantum error decoder designed for fault-tolerant quantum computers, along with its evaluation based on real chip measurements. The developed decoder supports code distance $d = 15$ and was fabricated using 22-nm CMOS technology, with measurement results confirming its expected operation. Building on this research, we plan to further refine the algorithm and microarchitecture, as well as explore ASIC implementations of other components within the back-end of fault-tolerant quantum computers.

ACKNOWLEDGMENT

This work was supported by JST Moonshot R&D Grant Number JPMJMS226A. This work was also supported through the activities of VDEC, The University of Tokyo, in collaboration with NIHON SYNOPSIS G.K and Cadence Design Systems.

REFERENCES

- [1] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," *Reviews of Modern Physics (Rev. Mod. Phys.)*, vol. 92, no. 1, p. 015003, March 2020.
- [2] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, "Practical quantum advantage in quantum simulation," *Nature*, vol. 607, no. 7920, pp. 667–676, July 2022.
- [3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A (Phys. Rev. A)*, vol. 86, no. 3, p. 032324, September 2012.
- [4] B. M. Terhal, "Quantum error correction for quantum memories," *Reviews of Modern Physics (Rev. Mod. Phys.)*, vol. 87, no. 2, pp. 307–346, April 2015.
- [5] N. Liyanage, Y. Wu, S. Tagare, and L. Zhong, "FPGA-Based Distributed Union-Find Decoder for Surface Codes," *IEEE Transactions on Quantum Engineering (TQE)*, vol. 5, pp. 1–18, September 2024.
- [6] B. Barber, K. M. Barnes, T. Bialas, O. Buğdaycı, E. T. Campbell, N. I. Gillespie, K. Johar, R. Rajan, A. W. Richardson, L. Skoric, C. Topal, M. L. Turner, and A. B. Ziad, "A real-time, scalable, fast and highly resource efficient decoder for a quantum computer," 2023. [Online]. Available: <https://arxiv.org/abs/2309.05558>
- [7] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, "Towards Practical Classical Processing for the Surface Code," *Physical Review Letters (Phys. Rev. Lett.)*, vol. 108, no. 18, p. 180501, May 2012.
- [8] K. M. Svore, D. P. Divincenzo, and B. M. Terhal, "Noise threshold for a fault-tolerant two-dimensional lattice architecture," *Quantum Information & Computation (QIC)*, vol. 7, no. 4, pp. 297–318, May 2007.

- [9] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, "Surface code quantum computing by lattice surgery," *New Journal of Physics (NJP)*, vol. 14, no. 12, p. 123011, December 2012.
- [10] D. Litinski, "A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery," *Quantum*, vol. 3, p. 128, March 2019.
- [11] A. G. Fowler, A. C. Whiteside, A. L. McInnes, and A. Rabbani, "Topological Code Autotune," *Physical Review X (Phys. Rev. X)*, vol. 2, p. 041003, October 2012.
- [12] S. Wøhlk and G. Laporte, "Computational comparison of several greedy algorithms for the minimum cost perfect matching problem on large graphs," *Computers & Operations Research (COR)*, vol. 87, pp. 107–113, November 2017.
- [13] N. Delfosse and N. H. Nickerson, "Almost-linear time decoding algorithm for topological codes," *Quantum*, vol. 5, p. 595, December 2021.
- [14] Y. Wu, N. Liyanage, and L. Zhong, "An interpretation of union-find decoder on weighted graphs," 2022. [Online]. Available: <https://arxiv.org/abs/2211.03288>
- [15] Y. Ueno, M. Kondo, M. Tanaka, Y. Suzuki, and Y. Tabuchi, "QECool: On-Line Quantum Error Correction with a Superconducting Decoder for Surface Code," in *ACM/IEEE Design Automation Conference (DAC)*, December 2021, pp. 451–456.
- [16] P. Das, C. A. Pattison, S. Manne, D. M. Carmean, K. M. Svore, M. Qureshi, and N. Delfosse, "AFS: Accurate, Fast, and Scalable Error-Decoding for Fault-Tolerant Quantum Computers," in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, April 2022, pp. 259–273.
- [17] W. Liao, Y. Suzuki, T. Tanimoto, Y. Ueno, and Y. Tokunaga, "WIT-Greedy: Hardware System Design of Weighted Iterative Greedy Decoder for Surface Code," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, January 2023, pp. 209–215.