一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

誤り耐性量子コンピュータに向けた22nmバルクプロセスによる表面符号用エラー訂正復号器の設計

青山　　連† 　門本淳一郎†† 　小林　和淑†

† 京都工芸繊維大学
†† 東京大学

あらまし　実用的な計算をできる量子コンピュータの実現には誤り訂正機能が必要である．表面符号は代表的な誤り訂正手法のひとつであり高い誤り訂正機能を持つ．本研究では誤り訂正機能の一部である復号器をハードウェア記述言語である Verilog HDL を用いて設計した．設計にあたって，貪欲法と呼ばれるアルゴリズムを使用した．設計した HDL を 22nm バルクプロセスで論理合成を行い，動作周波数，面積，消費電力を求めた．復号器内のノード数を変更することによる面積，消費電力を求めた．エントリ数 40 では 1GHz まで動作することを確認し，そのときの面積，消費電力はそれぞれ 5.85μm^2, 4.28mW であった．

キーワード　表面符号, 量子コンピュータ, 復号器, エラー訂正, 貪欲法

Error Correction Decoder of the Surface Code designed in a 22-nm Bulk Process for Fault Torelant Quantum Computers

Ren AOYAMA†, Junichiro KADOMOTO††, and Kazutoshi KOBAYASHI†

† Kyoto Institute of Technology
†† The University of Tokyo

Abstract　Error correction is mandatory to realize a quantum computer that can perform practical calculations. Surface codes is one of typical error correction methods because of their high error correction capability. In this study, we designed an error decoder, which is a part of the error correction function, using Verilog HDL. The greedy algorithm is used in the design. The designed HDL was synthesized in a 22-nm bulk process to evaluate performance, power and area. Its area and power consumption are also evaluated by changing the number of nodes that can be stored in the decoder. As a result, it is confirmed that the circuit can operate up to 1 GHz with 40 entries. The area and power consumption were 5.85μm^2 and 4.28mW, respectively.

Key words　Surface code, Quantum computer, Decoder, Error correction, Greedy algorithm

1. はじめに

現在の半導体技術はムーアの法則に従い微細化，発展してきた．しかし，微細化による性能向上にも限界が指摘されてきた．これに対してコンピュータの応用範囲は拡大し，求められる計算量も増加している．これに対応するための新たなコンピュータとして量子コンピュータが注目されている．量子コンピュータとは量子もつれや重ね合わせ状態など，量子力学的性質を利用したコンピュータである．その特性を利用することにより高速な計算を可能にするとされている．しかし，現在の量子コンピュータのエラー率は実用的な計算を行うには非常に高いため，エラー訂正機能が必須である．

エラー訂正手法として量子誤り訂正符号がある．その中でも量子ビットを 2 次元平面上に並べた表面符号は誤り訂正しきい値が高く注目されている [1]．表面符号におけるエラー訂正ではエラーを検出した量子ビットの位置の情報をもとに正しいエラー位置を推定する必要がある．その処理を行う部分を復号器と呼ばれ，研究が行われている [2], [3], [4].

本研究では貪欲法と呼ばれるアルゴリズムを使用した復号器をハードウェア記述言語である Verilog-HDL で設計し，論理合成を行い，面積，消費電力を求めた．第 2 節では量子誤り訂正について述べる．第 3 節では設計した復号器と使用したアルゴリズムについて述べる．第 4 節では論理合成を行った結果，得られた消費電力，面積について述べる．第 5 節で結論を述べる．

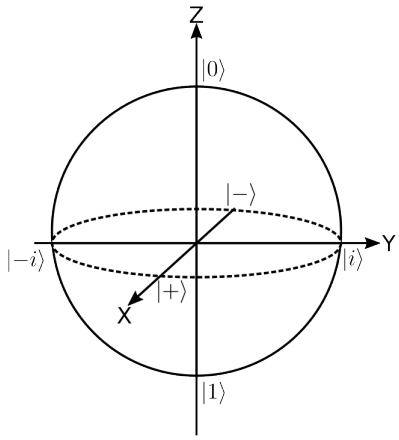

図 1　ブロッホ球

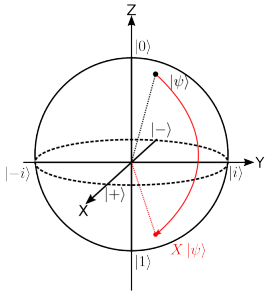

図 2　X エラー

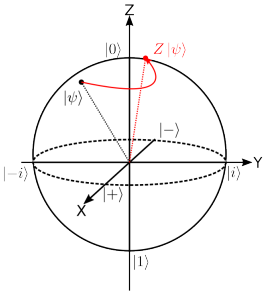

図 3　Z エラー

2.　量子エラー訂正

2.1　量子コンピュータにおけるエラー

量子コンピュータにおける情報の基本単位は量子ビットである．量子ビットの状態 $|\psi\rangle$ は二つの基底状態 $|0\rangle, |1\rangle$ と複素数 α, β を用いて $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ と表される．ただし，$|\alpha|^2 + |\beta|^2 = 1$ である．量子ビットの状態はブロッホ球を用いて表される．ブロッホ球を図 1 に示す．

量子コンピュータには古典コンピュータにもみられるビット反転エラー以外のエラーが存在する．X エラーは古典コンピュータと同様のビット反転エラーであり，量子ビットの状態は $X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle$ と表される．Z エラーは量子コンピュータ特有のもので，位相反転エラーと呼ばれ，$Z|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$ と表される．それぞれのエラーを図 2，3 に示す．この他に，Y エラーと呼ばれるものもあり，これは X エラーと Z エラーの組み合わせで表される．

2.2　表 面 符 号

表面符号は，物理量子ビットを 2 次元平面上に配置し 1 つの論理量子ビットを構成する誤り訂正符号である．表面符号の構成を図 4 に示す．符号を構成する物理量子ビットは情報を保持するデータ量子ビットと，データ量子ビットに発生したエラーを検知する補助量子ビットに分けられる．補助量子ビットは隣接したデータ量子ビットを観測し，エラーの有無を調べる [5]．これにより得られる測定値をエラーシンドロームと呼び，この情報をもとに訂正処理が行われる．量子ビットには 2 種類のエラーがあるため，補助量子ビットも 2 種類に分けられる．

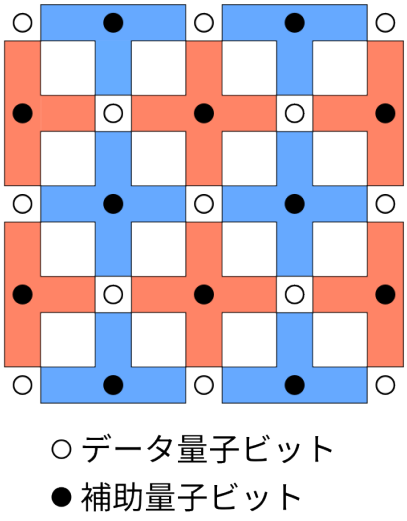

図 4　表面符号の概略図

表面符号の復号方法について述べる．補助量子ビットは隣接した奇数個のデータ量子ビットにエラーが発生した際，1 を示す．この 1 を示した補助量子ビットをアクティブノードと呼ぶ．量子コンピュータが計算している間，一定間隔毎に補助量子ビットの観測が行われる．この間隔をコードサイクルと呼ぶ．

表面符号における復号とは，補助量子ビットから得られるエラーシンドロームをもとにデータ量子ビットのエラー位置を推定し訂正することである．この推定するという問題には，近似的に解くことができる方法を用いることが一般的である．代表的なものとして，アクティブノードを頂点とし，アクティブノード間のマンハッタン距離を重みとしたグラフの最小重みマッチング問題として扱う手法である．復号器ではアクティブノード間のマンハッタン距離が最小となるようなアクティブノードの組を決定し，そのノード間にエラーが起きたデータ量子ビットが存在すると推定する．

エラーを推定する際，推定し訂正するエラーはデータ量子ビットのみではなく補助量子ビットの測定エラーについても行う必要がある．測定エラーについては，複数回補助量子ビットを観測することで対応できる．ある時間の観測結果とそのあとの観測結果，その二つの観測結果の XOR 値を重ねた 3 次元グラフ上でマッチング問題を解くことで測定エラーの検知，訂正を行うことができる．

3.　復号器の設計

3.1　使用アルゴリズム

今回設計した復号器では貪欲法と呼ばれるアルゴリズム [6] を用いた．貪欲法はスケーラビリティと小さなレイテンシを示すことが可能であり，先行研究でも用いられている [7], [3], [4]．ただし，得られる解が大域的最適解ではなく局所的最適解となる可能性がある．

先行研究 [7] では同様に貪欲法を用いた復号器が設計され

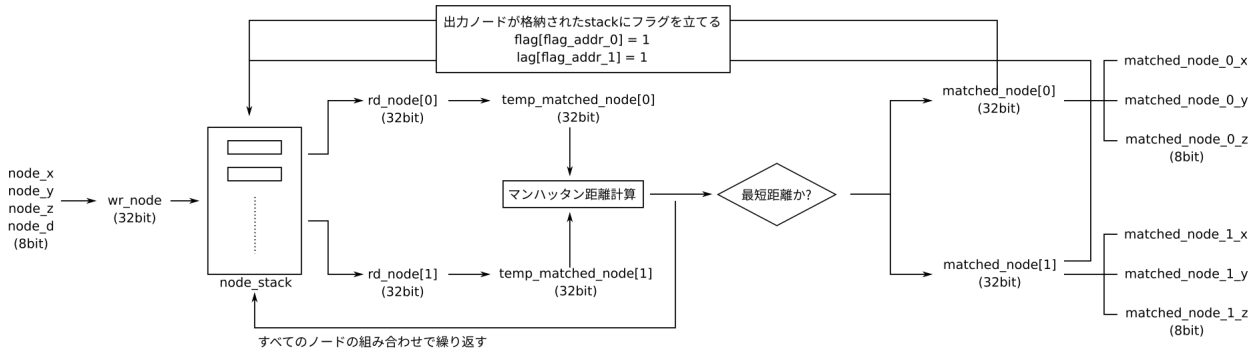

図 5　復号器の概略図

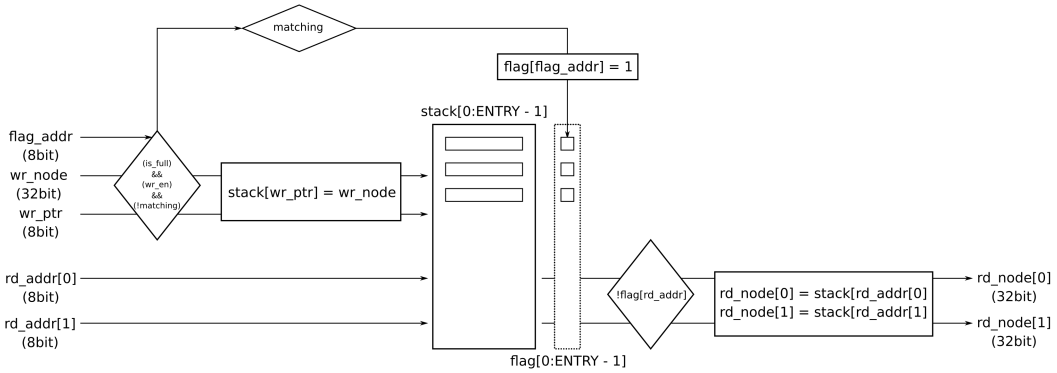

図 6　node_stack の概略図

ている．先行研究で用いられている貪欲法は iterative greedy algorythm と呼ばれる異なったアルゴリズムである．実装が容易で先に述べたように良い性質を持つため本研究では通常の貪欲法を用いている．

アルゴリズムの概要を以下に示す．

（1） ノードを 1 つ選択する．
（2） 選択したノードとそれ以外の全ノードとのマンハッタン距離を求める．
（3） 最も距離の短いノードの組を出力する．
（4） 2 つ目以降のノードにも同様の処理を行う．

3.2　復号器の構成

復号器の構成の概要を図 5 に示す．入力したノードを格納する node_stack とマンハッタン距離を計算し，最短距離となるノードの組を推定する decoder に分かれる．動作の概要を次に示す．

（1） アクティブノードの座標を入力とし，node_stack へ格納する．
（2） node_stack から 2 つのノードを読み出し，ノード間のマンハッタン距離を計算する．
（3） 手順 2 で求めたマンハッタン距離が最短であればそのノードの組を出力とする．
（4） 全ノードの組み合わせでこの処理を行い，最短距離となるノードの組を決定する．
（5） ノードが決定するとその情報を出力すると同時に，そのノードが格納されていた stack のアドレスにフラグを立てる．
（6） フラグを立てたアドレスの stack からはノードが読み出されない．
（7） これにより stack 内のフラグの立っていないアドレスからノードを読み出し，最短距離となるノードの組み合わせを求める．
（8） stack 内のノードすべての組が決まるまで処理を繰り返す．

マンハッタン距離の計算方法について述べる．ノード 0 の座標 (x_0, y_0, z_0)，境界までの距離 d_0，ノード 1 の座標 (x_1, y_1, z_1)，境界までの距離 d_1 とする．ノード間のマンハッタン距離 d_m と境界を超えたときのノード間の距離 d_b はそれぞれ次のように求められる．d_m と d_b のより短いほうが最終的なノード間の距離 d として記録される．

$$d_m = |x_0 - x_1| + |y_0 - y_1| + |z_0 - z_1|$$
$$d_b = d_0 + d_1$$
$$d = \min(d_m, d_b)$$

node_stck の概略図を図 6 に示す．node_stack は入力されたノードを格納する stack と各 stack のアドレスに対応する flag の 2 種類の配列からなる．格納可能なノードの数をエントリ数と

ソースコード 1　作成した Verilog HDL コードの一部

```
1   //マンハッタン距離の測定
2   assign manhattan_dist =
3             abs(temp_matched_node[0][31:24] -
                  temp_matched_node[1][31:24])
4             + abs(temp_matched_node[0][23:16] -
                  temp_matched_node[1][23:16])
5             + abs(temp_matched_node[0][15:8] -
                  temp_matched_node[1][15:8]);
6
7   assign temp_boundary_dist =
8   ((temp_matched_node[0][7:0] + temp_matched_node
        [1][7:0]) != 8'd0) ? temp_matched_node
        [0][7:0] + temp_matched_node[1][7:0] : 8'
        b11111111;
9
10  //絶対値化
11  function signed [7:0] abs (input signed [7:0] in);
12      if (in < 0) begin
13          abs = - in;
14      end else begin
15          abs = in;
16      end
17  endfunction
```

呼ぶ．flag が 1 となっているアドレスの stack からは読み出しが行われない．

設計した Verilog HDL コードの一部をソースコード 1 に示す．示したのはマンハッタン距離を計算する部分である．manhattan_dist がノード間の距離，temp_boundary_dist が境界をまたいだ場合のノード間の距離，temp_matched_node は距離を求めるノードである．MSB から 8bit 毎にそれぞれ x 座標，y 座標，z 座標，境界までの距離を表す．

4. 合成結果

22nm バルクプロセスで，論理合成を行った．クロック周波数を変更し，動作可能かを確認し，その周波数における面積，消費電力について調べた．文献 [8] によると，物理量子ビットエラー率 $p = 10^{-4}$，符号距離 $d = 15$，論理量子ビットエラー率 $p_L = 10^{-15}$ で，エントリ数 30, $p = 10^{-3}, d = 31, p_L = 10^{-15}$ のとき，エントリ数が 70 あれば十分であると推定されている．そのため，エントリ数を 40 と 80 の 2 通りでそれぞれの面積，消費電力も求めた．比較のため ASAP の 7nmFinFET ライブラリを用いて論理合成を行った [9].

合成して得られた結果を表 1，2 に示す．表 1 では動作周波数，エントリ数の違いによる面積の変化を示している．エントリ数 40 と 80 のときでは面積に約 2 倍となる．今回設計した復号器全体の中で node_stack 内の stack が 32bit で最もビット数が大きいため面積を占める．その大きさが 2 倍になるためこのような結果となる．表 2 では動作周波数による変化，エントリ数による消費電力の変化を示している．周波数による消費電力の増加はエントリ数 40 のときに 391%，80 のときに 392% であった．

表 1　22nm バルクプロセスでのエントリ数の変化による面積の変化

	面積　($\times 10^3$)	
周波数	エントリ数 40	エントリ数 80
200MHz	$5.73\mu m^2$	$10.1\mu m^2$
500MHz	$5.73\mu m^2$	$10.1\mu m^2$
1GHz	$5.85\mu m^2$	$10.2\mu m^2$

表 2　22nm バルクプロセスでの動作速度の変化による消費電力の変化

	消費電力	
周波数	エントリ数 40	エントリ数 80
200MHz	0.870 mW	1.53 mW
500MHz	2.15 mW	3.78 mW
1GHz	4.28 mW	7.54 mW

表 3　7nmFinFET プロセスでのエントリ数の変化による面積の変化

	面積　($\times 10^3$)	
周波数	エントリ数 40	エントリ数 80
200MHz	$1.15\mu m^2$	$2.05\mu m^2$
500MHz	$1.15\mu m^2$	$2.05\mu m^2$
1GHz	$1.16\mu m^2$	$2.06\mu m^2$
1.5GHz	$1.12\mu m^2$	$2.02\mu m^2$

表 4　7nmFinFET プロセスでの動作速度の変化による消費電力の変化

	消費電力	
周波数	エントリ数 40	エントリ数 80
200MHz	0.466 mW	0.827 mW
500MHz	1.15 mW	2.04 mW
1GHz	2.29 mW	4.06 mW
1.5GHz	3.33 mW	5.99 mW

22nm バルクプロセスでは 1GHz が動作する最大周波数であった．そのときの面積，消費電力はそれぞれエントリ数 40 のとき，$5.85\mu m^2$, 4.28mW，80 のとき，$10.2 \times 10^3 \mu m^2$, 7.54mW であった．

7nmFinFET ライブラリを用いて合成した結果を表 3，4 に示す．1.5GHz まで動くことを確認し，そのときの面積，消費電力はそれぞれエントリ数 40 で $1.12 \times 10^3 \mu m^2$, 3.33mW，80 では $2.02 \times 10^3 \mu m^2$, 5.99mW となる．

22nm バルクプロセスにおける最大動作周波数である 1GHz においてのそれぞれの面積，消費電力について述べる．エントリ数 40 では 7nmFinFET プロセスでの面積，消費電力はそれぞれ 22nm バルクプロセスの結果の 19.8%, 53.5% であった．エントリ数 80 のときはそれぞれ，19.8%, 53.8% となった．

5. おわりに

本稿では，ハードウェア記述言語である Verilog を用いて，貪欲法を使用した復号器を設計した．22nm バルクプロセスで論理合成を行い，消費電力，面積の確認を行った．また，エントリ数を変化させたときの電力と面積も確認した．

動作する最大周波数は 1GHz であり，そのときの面積，消費電力はエントリ数 40 のとき，$5.85\mu m^2$, 4.28mW となり，80 のときには $2.02 \times 10^3 \mu m^2$, 5.99mW となった．

文　　献

[1] A.G. Fowler, M. Mariantoni, J.M. Martinis, and A.N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” Phys. Rev. A, vol.86, p.032324, Sept. 2012. https://link.aps.org/doi/10.1103/PhysRevA.86.032324

[2] P. Das, C.A. Pattison, S. Manne, D. Carmean, K. Svore, M. Qureshi, and N. Delfosse, “A scalable decoder micro-architecture for fault-tolerant quantum computing,” arXiv preprint arXiv:2001.06598, pp.••–••, 2020.

[3] Y. Ueno, M. Kondo, M. Tanaka, Y. Suzuki, and Y. Tabuchi, “Qecool: On-line quantum error correction with a superconducting decoder for surface code,” 2021 58th ACM/IEEE Design Automation Conference (DAC)IEEE, pp.451–456 2021.

[4] W. Liao, Y. Suzuki, T. Tanimoto, Y. Ueno, and Y. Tokunaga, “Wit-greedy: Hardware system design of weighted iterative greedy decoder for surface code,” Proceedings of the 28th Asia and South Pacific Design Automation Conference, pp.209–215, 2023.

[5] 徳永裕己他，“量子コンピュータ: 3. 量子コンピュータの誤り訂正技術-物理に即したトポロジカル表面符号，” 情報処理，vol.55，no.7，pp.695–701，2014.

[6] D.E. Drake and S. Hougardy, “A simple approximation algorithm for the weighted matching problem,” Information Processing Letters, vol.85, no.4, pp.211–213, 2003. https://www.sciencedirect.com/science/article/pii/S0020019002003939

[7] 中村徹舟，宮村信，井上弘士，川上哲志，阪本利司，多田宗弘，谷本輝夫他，“極低温不揮発 fpga を対象とした誤り耐性量子コンピュータ向け表面符号復号器の rtl 設計，” 研究報告システムと LSI の設計技術 (SLDM)，vol.2023，no.27，pp.1–10，2023.

[8] Y. Suzuki, T. Sugiyama, T. Arai, W. Liao, K. Inoue, and T. Tanimoto, “Q3de: A fault-tolerant quantum computer architecture for multi-bit burst errors by cosmic rays,” 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), pp.1110–1125, 2022.

[9] L.T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, “Asap7: A 7-nm finfet predictive process design kit,” Microelectronics Journal, vol.53, pp.105–115, 2016. https://www.sciencedirect.com/science/article/pii/S002626921630026X