Implementing Logical Operators using Code Rewiring

Darren Banfield and Alastair Kay

Department of Mathematics, Royal Holloway University of London, Egham Hill, Egham TW20 0EX, UK.

We describe a method to use measurements and correction operations in order to implement the Clifford group in a stabilizer code, generalising a result from [1] for topological subsystem colour codes. In subsystem stabilizer codes of distance at least 3 the process can be implemented fault-tolerantly. In particular this provides a method to implement a logical Hadamard-type gate within the 15-qubit Reed-Muller quantum code by measuring and correcting only three observables. This is an alternative to the method proposed by [2] to generate a set of gates which is universal for quantum computing for this code. The construction is inspired by the description of code rewiring from [3], and may have some application to quantum low density parity check codes.

1 Introduction

Quantum computation offers the prospect of a new approach to computing, with the potential to be much more efficient than classical computers for certain problems (as exemplified in [4] and [5]). However, in practice a major limitation of current quantum computing hardware is its susceptibility to various physical sources of errors. These errors create noise in the calculations which, if left unchecked, can affect and damage the output of any computation. Much current research is aimed at correcting for these errors as they occur, which should allow experimentalists to implement large-scale reliable computations and help quantum computers achieve their potential (see for instance [6–8]).

Research has focused on overcoming the errors by using a quantum error-correcting code to encode the data from a single ("logical") qubit in a block consisting of multiple physical qubits. Quantum gates are performed on these blocks of qubits, in such a way as to induce the desired logical operation on the corresponding logical qubit. If these gates are constructed carefully then, by repeatedly encoding blocks of physical qubits, circuits can be designed to achieve any desired accuracy as long as the physical error rate is below an error threshold. Such a construction is said to be fault-tolerant. Given the prevalence of errors, it is desirable that this threshold be as high as possible; one active area of research is in developing codes which have a high threshold [9, 10].

If all logical gates could be implemented transversally, so that each qubit interacts only with its corresponding qubit in any block, then fault-tolerance would be easily achieved. It is also likely that transversal gates would lead to a high error threshold. However, in [11] Eastin and Knill show that it is not possible to construct a universal gate set for a quantum error-correcting code using only transversal unitary gates.

Various authors ([1, 3, 12–15]) have developed ways to circumvent this issue, by using two codes which between them permit transversal implementations of a universal set of gates. Switching between these two codes allows transversal gates to be used at all times. These switches may be achieved by a process called code rewiring (or code deformation) which involves repeatedly performing a measurement followed by a correction based upon the measurement outcome. Each round of measurement and correction can change the code, and the aim of the process is to switch between two given codes.

The code switching process can induce a logical operation on the code space. Colladay and Mueller [3] proved that these operations are Clifford. Here, by restricting to the case where the initial and final codes are the same, we investigate which logical operators can be induced by rewiring. We show in Section 3 that the whole Clifford group can be induced in this way. Furthermore we show in Section 4 that if a code results from a non-trivial gauge fixing of a subsys-

Darren Banfield: darren.banfield.2020@live.rhul.ac.uk
Alastair Kay: alastair.kay@rhul.ac.uk

tem code with distance d, then these operators can be induced using intermediate codes whose distance is at least d.

This result is particularly interesting for codes that do not have transversal Clifford gates. In particular, our result applies to the 15-qubit Reed-Muller code and we show that the Clifford group can be generated fault-tolerantly using only measurements and corrections in this code. It is shown in [16] that this is the smallest stabilizer code which admits a transversal implementation of the T gate. This provides an alternative method to [2] to generate a set of gates which is universal for quantum computation for the Reed-Muller code, since the only additional gate required for universality is T, which is transversal.

Recently, there has been much interest in quantum low density parity check codes (LDPC codes - see [17] for an overview). We provide some comments in Section 5 on the applicability of this code rewiring approach to logical gates within these quantum LDPC codes.

1.1 Notation

Suppose that C is a code in the Hilbert space $\mathcal{H}$ of n physical qubits. If C has k logical qubits with a code distance of d then we say that C is an $[[n, k, d]]$-code.

Given an element $\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{F}_2^n$ we will write X_λ to denote the operator

$$X_\lambda = \prod_{j:\lambda_j=1} X_j. \tag{1}$$

That is, X_λ is the operator which acts as the Pauli X matrix on physical qubits j for which $\lambda_j = 1$. We define the operator Z_λ similarly.

Suppose that a $[[n, k, d]]$ quantum stabilizer code C has check matrix $M = \begin{pmatrix} M_X & M_Z \end{pmatrix}$, with logical operators

$$L_X^{(j)} = X_{L_{X^{(j)},X}} Z_{L_{X^{(j)},Z}}$$
$$L_Z^{(j)} = X_{L_{Z^{(j)},X}} Z_{L_{Z^{(j)},Z}}$$

for $j = 1, \dots, k$. Furthermore let α_1 and α_2 be two elements of $\mathbb{F}_2^n$ and $\alpha = (\alpha_1, \alpha_2)$. Then we will write $\Lambda(M, \alpha, L_X, L_Z)$ to denote the matrix

$$\begin{pmatrix} M_X & M_Z \\ \alpha_1 & \alpha_2 \\ L_{X^{(1)},X} & L_{X^{(1)},Z} \\ L_{Z^{(1)},X} & L_{Z^{(1)},Z} \\ \vdots & \vdots \\ L_{X^{(k)},X} & L_{X^{(k)},Z} \\ L_{Z^{(k)},X} & L_{Z^{(k)},Z} \end{pmatrix}. \tag{2}$$

If the α component is missing then $\Lambda(M, L_X, L_Z)$ will denote the same matrix, but with the row relating to α removed. Taken in context this should be unambiguous.

We will have occasion to consider the commutativity relationships between an observable and the logical operators of a code. Suppose that g is member of the Pauli group on n qubits which is an observable, and that U is a logical operator on C (also a member of the Pauli group). Two members of the Pauli group either commute or anti-commute, and we define $c(g, U)$ such that

$$gU = (-1)^{c(g,U)} Ug. \tag{3}$$

That is, $c(g, U) = 0$ if U and g commute, and 1 otherwise.

We are able to prove stronger results in the case that the stabilizer code is also a subsystem code. We can consider the stabilizer code as an $[[n, k, d]]$ stabilizer code for some $k > 1$, but with a subset of the k logical qubits fixed. The result is an $[[n, k', d']]$ stabilizer code which has fewer logical qubits, but the same or greater code distance. Selecting different subsets of the k logical qubits and fixing these by adding additional stabilizers can result in various distinct $[[n, k', d']]$ stabilizer codes for $k' < k$ and $d' \geq d$. For more details on subsystem stabilizer codes, see [18].

2 Code Rewiring

The principle behind code rewiring is established in Example 9 of [19], which we reformulate as the following statement.

Lemma 1. *Let C be a quantum stabilizer code on the n-qubit state space $\mathcal{H}$, and $g_1, \dots, g_m$ a set of generators for its stabilizer group S. Let $g \in G_n$, the Pauli group on $\mathcal{H}$, be an observable which anticommutes with g_m. Suppose that we measure g and apply the operator g_m if the measurement* *outcome is -1. Then the stabilizer generators and logical operators evolve as follows:*

- *g_m is removed from the set of generators and replaced with g.*
- *For every operator h where h is either another generator of S or a logical operator, h transforms to $g_m^{c(g,h)}h$.*

It is shown in [3] that the procedure in Lemma 1 has the net effect of applying the operator $U = (I + gg_m)/\sqrt{2}$ to the state space; moreover it is shown in Appendix A of [3] that U is a Clifford operator.

In [3], Colladay and Mueller use this procedure to provide an algorithm to convert a state encoded in one stabilizer code C into a state encoded in a different stabilizer code C'. Given a set S of generators $g_1, \ldots, g_m$ for code C and a set S' of generators $g'_1, \ldots, g'_m$ for code C', the authors find a sequence of measurements which have the effect of converting each g_j into a corresponding g'_j. They provide an algorithm for determining appropriate measurements.

In particular, in the context of this paper, they apply this to map between the Steane code and the 15-qubit Quantum Reed-Muller code $QRM(4)$. Earlier work in [12] had also provided a map between these codes (and adjacent Quantum Reed-Muller codes more generally) by considering both codes as different gauge-fixings of a larger code. This is similar to the approach taken in [2].

Colladay and Mueller leave open the question of which Clifford operators can be implemented such a sequence of measurements and error corrections. We will show that any Clifford operator can be implemented in this way.

Note that it is sufficient to consider the case in which $C = C'$. Suppose that any Clifford operator can be induced by this method in the case in which the original code and final code are the same. Then from [3], there is a measurement process mapping from C to C' which induces some Clifford operator V, so the Clifford operator $U : C \to C'$ can be induced by inducing the Clifford operator $UV^\dagger$ on C'.

3 Generating the Clifford Group

We start by considering the case of a single logical qubit. In section 3.3 we will extend to the multiple qubit case.

Definition 1. Let C be a stabilizer code which has stabilizer generators $g_1, g_2, \ldots, g_m$. We say that a pair (g, g') of observables is a code rewiring pair if for one of the stabilizer generators g_m,

1. $c(g, g_j) = 0$ for $j = 1, 2, \ldots, m-1$ and $c(g, g_m) = 1$ (that is, g commutes with all of the stabilizer generators except for g_m, with which it anticommutes),
2. $c(g, X_L) = 0 = c(g, Z_L)$,
3. $c(g', g_j) = 0$ for $j = 1, 2, \ldots, m-1$ and $c(g', g_m) = 1$, and
4. $c(g, g') = 1$.

Our approach to implementing non-trivial Clifford operators on the code space C involves choosing a stabilizer generator g_m and a corresponding code rewiring pair (g, g'). Note that the second condition in this definition is not necessary, but it simplifies the discussion and imposing this condition has no effect on the generality of our results. (If more than one generator anticommutes, one can always redefine the generating set such that there is only one anti-commuting member.)

Definition 2. Given a code rewiring pair, an elementary code rewiring consists of the following set of measurements and corrections:

- Measure g and apply g_m if the measurement outcome is -1.
- Measure g' and apply g if the measurement outcome is -1.
- Measure g_m and apply g' if the measurement outcome is -1.

Note that under this procedure, the stabilizer g_m is replaced by g in the first step, since $\{g_m, g\} = 0$. In the second step, because $\{g, g'\} = 0$, the new stabilizer g is replaced by g'. In the final step, g' is replaced by g_m. Thus overall the observable g_m evolves through the elementary code rewiring as

$$g_m \mapsto g \mapsto g' \mapsto g_m.$$

The rest of the stabilizer generators are unchanged, so that the code C is mapped to itself. This three-step process therefore maps the codespace to itself and hence induces a logical operator on C (which is shown in [3] to be a Clifford operator).

Note that at least three steps are required to produce a non-trivial operator. Suppose instead we were to apply the process of Lemma 1 to only two measurements, of g and g_m. Then condition 2 of Definition 1 implies that the induced operator is trivial. Relaxing condition 2, so that say g and X_L anticommute, would mean that X_L evolves into $g_m X_L = X_L$ under the first operation. But g_m commutes with X_L and so X_L is unaffected by the second measurement, and the induced operation is trivial.

3.1 Logical Operators

The transformation of the logical operators under an elementary code rewiring is determined by the commutativity relationships between them and g'. Let us write X'_L and Z'_L for the images of the logical operators X_L and Z_L after the elementary code rewiring. If g' commutes with both X_L and Z_L then these operators are not affected by the procedure, so $X'_L = X_L$, $Z'_L = Z_L$, and the identity operator has been applied to C.

However, suppose that $[g', X_L] = 0$ while $\{g', Z_L\} = 0$. Then when the measurement of g' is made, Z_L is replaced by gZ_L. Observe that Z_L commutes with g_m since g_m is a stabilizer of C, but g anticommutes with g_m. Hence, when the final measurement (that of g_m) is made, the operator gZ_L is replaced by $g'gZ_L$. Overall the logical Z operator evolves as

$$Z_L \mapsto Z_L \mapsto gZ_L \mapsto Z'_L := g'gZ_L.$$

Now Z_L anticommutes with X_L, while both g' and g commute with X_L so Z'_L anticommutes with X_L. Further, Z_L commutes with itself and with g but anticommutes with g' and therefore Z'_L anticommutes with Z_L. From this we conclude that, as the induced map is a Clifford operator, $Z'_L = \pm Y_L$. Since X_L commutes with every observable measured, it is unchanged by the process. The elementary code rewiring thus maps

$$X_L \mapsto X_L$$
$$Z_L \mapsto \pm Y_L$$

and therefore the induced operation on C is either $\sqrt{X}$ or $\sqrt{X}^\dagger = X\sqrt{X}$ depending upon the sign above.

Commutativity $c(g', X_L)$	Commutativity $c(g', Z_L)$	Image of X_L	Image of Z_L	Induced type
0	0	X_L	Z_L	identity
0	1	X_L	$\pm Y_L$	$\sqrt{X}$
1	0	$\pm Y_L$	Z_L	$\sqrt{Z}$
1	1	$\pm Z_L$	$\pm X_L$	$\sqrt{Y}$

Table 1: Logical operators induced by the code rewiring process. Here, $c(g', X_L)$ and $c(g', Z_L)$ denote commutation of g' and the relevant logical operator, as defined in Equation (3).

To simplify notation, we make the following definition.

Definition 3. For any $t \in \{X, Y, Z\}$, a logical operator on a code C encoding one logical qubit is said to be $\sqrt{t}$-type if it is either $\sqrt{t}$ or $t\sqrt{t}$.

The example above induces a $\sqrt{X}$-type operation on C.

Repeating the analysis above with the remaining two commutativity relationships gives the middle two columns of Table 1. From these columns we can deduce the final column, except for the bottom entry: the analysis does not rule out the possibility that the induced operator maps X_L to λZ_L and Z_L to λX_L for some $\lambda \in \{\pm 1\}$. However, if this were the case then

$$\lambda^2 X_L = \lambda g' g Z_L = g' g g' g X_L = -X_L$$

which does not allow $\lambda \in \{\pm 1\}$. Hence we conclude that if $c(g', X_L) = c(g', Z_L) = 1$ then the induced operation is of type $\sqrt{Y}$.

The table shows that, depending on the commutativity relationship between g', X_L and Z_L, an operator of any type $\sqrt{X}$, $\sqrt{Z}$ or $\sqrt{Y}$ can be induced on C. Since our aim is to induce non-trivial operators on C, we seek code rewiring pairs for which g' has each of the commutativity relationships with X_L and Z_L.

3.2 Single Logical Qubit

We have shown that if a suitable code rewiring pair can be found, we can create a range of single-qubit Clifford operations. We now need to prove that such a pair always exists.

Proposition 2. *Let C be a stabilizer code encoding one logical qubit. Then for any given $t \in \{\sqrt{X}, \sqrt{Y}, \sqrt{Z}\}$ there exists an elementary code rewiring such that the induced operator on C is of t-type.*

Proof. We use the notation defined in Section 1.1: the code C has check matrix M and logical operators $X_L = X_{L_{X,X}} Z_{L_{X,Z}}$, and Z_L, as defined in Equation (1).

Consider the following equation over $\mathbb{F}_2$ (recalling the notation from Equation (2):

$$\Lambda(M, L_X, L_Z) \begin{pmatrix} \alpha_Z^T \\ \alpha_X^T \end{pmatrix} = (0, \dots, 0, 1, 0, 0)^T. \quad (4)$$

The matrix has full rank because the stabilizer generators are independent and the logical operators do not lie in the stabilizer group. So the map $\mathbb{F}_2^{2n} \to \mathbb{F}_2^{n+1}$ is surjective and there are solutions to this equation. If $(\alpha_Z, \alpha_X)^T$ is one such solution then let

$$g = i^{|\alpha_X \cdot \alpha_Z|} X_{\alpha_X} Z_{\alpha_Z}$$

where the inner product in the factor is taken over $\mathbb{F}_2$. By construction g commutes with the logical operators X_L and Z_L, and with all stabilizer generators except for the final one in the representation M.

Now let $\alpha = (\alpha_X, \alpha_Z)$ and let $(a, b) \in \{0, 1\}^2$. Consider the equation

$$\Lambda(M, \alpha, L_X, L_Z) \begin{pmatrix} \beta_Z^T \\ \beta_X^T \end{pmatrix} = (0, \dots, 0, 1, 1, a, b)^T. \quad (5)$$

Because $X_{\alpha_X} Z_{\alpha_Z}$ anticommutes with the stabilizer g_m, this matrix also has independent rows and has full rank $n + 2$. Therefore just as for Equation (4) there are solutions to Equation (5). If $(\beta_Z, \beta_X)^T$ is one such solution then let

$$g' = i^{|\beta_X \cdot \beta_Z|} X_{\beta_X} Z_{\beta_Z}.$$

As above, g' commutes with every stabilizer generator except for the final one. It anticommutes with g, and its commutativity relationship with X_L and Z_L is determined by the pair (a, b), the elements of the first two columns of Table 1:

$$c(g', X_L) = a, \qquad c(g', Z_L) = b.$$

Since there exist solutions to Equations (4) and (5) for every pair (a, b), the rewiring pair (g, g') can be chosen to induce an operator of t-type for any $t = \sqrt{X}, \sqrt{Z}$ or $\sqrt{Y}$. □

From the above analysis it is not possible to determine whether the code rewiring pair (g, g') induces the operator $\sqrt{t}$ or $t\sqrt{t}$: explicit calculation is needed. However, a simple observation leads to the following.

Corollary 3. *Let C be a stabilizer code encoding one logical qubit. Then given any choice of operator $W \in \{\sqrt{X}, \sqrt{Y}, \sqrt{Z}\}$ there exists an elementary code rewiring such that the induced operator on C is W.*

Proof. Suppose that the code rewiring pair (g, g') induces an operator of $\sqrt{t}$-type for some $t \in \{X, Y, Z\}$ and consider the effect of replacing g' by $-g'$. This does not affect the commutativity relationships and so $(g, -g')$ is a code rewiring pair. If $c(-g', U) = 0$ for some logical operator U then U is unaffected by the elementary code rewiring, but if $c(-g', U) = 1$ then U evolves into $-g'gU$. Table 1 shows that this is equivalent to multiplying the operator induced by the pair (g, g') by a factor of t. Therefore if the code rewiring pair (g, g') incudes the operator $t\sqrt{t}$, then the pair $(g, -g')$ induces the operator $\sqrt{t}$. □

3.3 Multiple Logical Qubits

We have proven our result for a single qubit. We now generalise it to the case in which more than one qubit is encoded.

Suppose now that C is a $[[n, k, d]]$-code where $k > 1$, with logical operators $X_L^{(j)}$ and $Z_L^{(j)}$ for $j = 1, 2, \dots, k$ (using the notation of Section 1.1). Consider the following equation over $\mathbb{F}_2$:

$$\Lambda(M, L_X, L_Z) \begin{pmatrix} \alpha_Z^T \\ \alpha_X^T \end{pmatrix} = (0, \dots, 0, 1, 0, \dots, 0)^T \quad (6)$$

where the 1 on the right-hand side appears in position $n - k$. The $n - k + 2k = n + k$ rows of the matrix on the left-hand side are linearly independent and so the matrix is of full rank $n + k < 2n$; this means that there is a solution to Equation (6).

As before, for a solution $(\alpha_Z, \alpha_X)^T$, we write $\alpha = (\alpha_X, \alpha_Z)$ and define

$$g = i^{|\alpha_X \cdot \alpha_Z|} X_{\alpha_X} Z_{\alpha_Z}.$$

By construction g commutes with all logical operators, and all stabilizers except for g_{n-k}.

Since $X_{\alpha_X} Z_{\alpha_Z}$ anticommutes with the stabilizer g_{n-k}, the row (α_X, α_Z) does not lie in the span of the rows of the matrix in the left-hand side of Equation (6). Therefore for any

$(a_1, b_1, \ldots, a_k, b_k) \in \{0,1\}^{2k}$ there is a solution over $\mathbb{F}_2$ to the equation

$$\Lambda(M, \alpha, L_X, L_Z) \begin{pmatrix} \beta_Z^T \\ \beta_X^T \end{pmatrix} = (0, \ldots, 0, 1, 1, a_1, b_1, \ldots, a_k, b_k)^T. \quad (7)$$

Recalling the notation from Equation (3), the operator $g' = i^{|\beta_X \cdot \beta_Z|} X_{\beta_X} Z_{\beta_Z}$ satisfies

$$c(g', X_L^{(j)}) = a_j \qquad c(g', Z_L^{(j)}) = b_j$$

and g' commutes with all stabilizers except for g_{n-k} with which is anticommutes; g' also anticommutes with g. This observation brings us to our main result.

Theorem 4. *Let C be an $[[n,k,d]]$ stabilizer code. Then any Clifford operator on C can be induced by a sequence of elementary code rewirings.*

Proof. Notice that if $a_j = b_j = 0$ for some j then $X_L^{(j)}$ and $Z_L^{(j)}$ commute with both g and g', and hence are unaffected by the elementary code rewiring. Proposition 2 therefore shows that for any j we can obtain the logical operators $X_L^{(j)}$ and $Z_L^{(j)}$ (by applying the operators of type $\sqrt{X}$ and $\sqrt{Z}$ twice) as well as either $H^{(j)} X_L^{(j)}$ or $H^{(j)} Z_L^{(j)}$. We can thus implement the Hadamard gate H on any given logical qubit j. It is shown in [20] that the Clifford group can be generated by the H, S and cNOT gates.

It remains to show that we can implement a cNOT gate between any two logical qubits, with control qubit j and target qubit ℓ.

Let $a_j = b_\ell = 1$ and $b_j = a_\ell = 0$, and set $a_s = b_s = 0$ for all $s \neq j, \ell$. The logical operators evolve as follows

$$\begin{aligned} X_L^{(j)} &\mapsto X_L^{(j)\prime} := g' g X_L^{(j)} \\ Z_L^{(j)} &\mapsto Z_L^{(j)} \\ X_L^{(\ell)} &\mapsto X_L^{(\ell)} \\ Z_L^{(\ell)} &\mapsto Z_L^{(\ell)\prime} := g' g Z_L^{(\ell)}. \end{aligned}$$

Since $X_L^{(j)}$ commutes with itself and g but anticommutes with g' we have

$$\{X_L^{(j)\prime}, X_L^{(j)}\} = 0.$$

Furthermore $Z_L^{(1)}$ commutes with g and g' but anticommutes with $X_L^{(j)}$, and hence

$$\{X_L^{(j)\prime}, Z_L^{(j)}\} = 0.$$

This is the same as in the case of a single logical qubit. However now in addition $Z_L^{(\ell)}$ anticommutes with g' and commutes with both g and $X_L^{(j)}$, so that

$$\{X_L^{(j)\prime}, Z_L^{(\ell)}\} = 0.$$

Because $X_L^{(\ell)}$ commutes with each of g, g' and $X_L^{(j)}$ we have $[X_L^{(j)\prime}, X_L^{(\ell)}] = 0$. Overall this means that

$$X_L^{(j)} \mapsto X_L^{(j)\prime} = \pm Y_L^{(j)} X_L^{(\ell)}.$$

Similarly

$$Z_L^{(\ell)} \mapsto Z_L^{(\ell)\prime} = \pm Z_L^{(j)} Y_L^{(\ell)}.$$

As required for a cNOT gate controlled by logical qubit j, the $X_L^{(j)}$ has been copied onto qubit ℓ and Z_L^{ℓ} has been copied onto qubit j.

By applying logical $S = \sqrt{Z}$ or $S^\dagger$ as appropriate to logical qubit j, and logical $\sqrt{X}$ or $\sqrt{X}^\dagger$ as appropriate to qubit ℓ, the overall evolution of the operators becomes

$$\begin{aligned} X_L^{(j)} &\mapsto X_L^{(j)} X_L^{(k)} \\ Z_L^{(j)} &\mapsto Z_L^{(j)} \\ X_L^{(\ell)} &\mapsto X_L^{(\ell)} \\ Z_L^{(\ell)} &\mapsto Z_L^{(j)} Z_L^{(\ell)} \end{aligned}$$

and cNOT$(j \to \ell)$ has been implemented.

Together with $\sqrt{X}$, $\sqrt{Y}$ and $\sqrt{Z}$ this is sufficient to generate the Clifford group. □

This result generalises that of [1] which shows that any Clifford operator can be implemented using code deformation measurements within a topological subsystem colour code.

4 Subsystem Codes

In seeking a fault-tolerant implementation of this procedure, we make some observations:

- Given an appropriate code distance, the measurement and error correction steps can be implemented fault-tolerantly (see for instance [21]).
- However, at least one code involved must be a non-CSS code, even if the original code is a CSS code (this can be seen by considering the form of the solutions to the linear equations).

- It is not, in general, easy to determine the distance of the intermediate codes. They are frequently lower than the distance of the original code.

However, in the case of a subsystem stabilizer code this final point can be overcome, and a fault tolerant implementation is possible. In a subsystem code (see for instance [18]) we may write the Hilbert state space as $\widehat{C} \oplus \widehat{C}^{\perp}$ where $\widehat{C} = C \otimes B$. The stabilizers of C are those of $\widehat{C}$ together with the set of operators $I_C \otimes g_B$ where g_B is an operator acting only on the subsystem B. The code C is obtained from $\widehat{C}$ by fixing the gauge qubits, which is to say treating the logical operators on B as stabilizers of C.

Theorem 5. *Suppose that $\widehat{C}$ is a subsystem stabilizer code with $r \geq 1$ gauge qubits and with distance d, and that C is obtained from $\widehat{C}$ by fixing at least one gauge qubit. Then any Clifford operator on C may be implemented using a sequence of elementary code rewirings such that the intermediate codes all have distance at least d.*

Proof. Since $\widehat{C}$ has distance d, every subsystem of $\widehat{C}$ also has distance at least d.

We can write $\widehat{C} = C \otimes B$ for some subsystem B of gauge qubits. The logical operators on B which fix C are the gauge operators, and are stabilizers of C. In particular we may choose g_m to be one of these gauge operators. Then for any Clifford operator W, Theorem 4 says that there is a sequence of elementary code rewirings which induces the operator W on C. Each of the intermediate codes arises from fixing different gauges within the subsystem code $\widehat{C}$ and therefore has distance at least d. □

In particular the 15-qubit Reed-Muller code $QRM(4)$ can be viewed as a subsystem arising from fixing six gauge qubits of a $[[15, 7, 3]]$-code, as in [12, 15]. Thus Theorem 5 provides a method to implement an operator of $\sqrt{Y}$-type using three measurements and such that every intermediate code has distance at least 3. This method can therefore be implemented fault-tolerantly, providing the one extra gate required to elevate the transversal gates to universal status.

5 Discussion

We have shown that code rewiring can be used to generate any element of the Clifford group for a stabilizer code C, and that under some mild conditions this can be carried out fault-tolerantly. This answers a question raised in [3]. While our approach is efficient for certain non-trivial gates, involving only elementary code rewirings, it is unlikely that it will in general be the most efficient. Combining two elementary code rewirings involves returning back to the original code as an intermediate step, and it is most likely that an alternative set of rewirings would involve fewer steps. It would be of interest to determine the most efficient set of measurements. One simple improvement would be to classically track the measurement outcomes, and apply a single operator after all measurements are made.

Our approach provides a method to generate a set of gates which is universal for quantum calculations in the 15-qubit quantum Reed-Muller code $QRM(4)$, which is the smallest code to have a transversal T gate [16], using just three rounds of measurements to implement non-basis preserving Clifford operators, supplemented by the transversal T gate. In contrast to the method in [2] in which the authors implement the Hadamard gate in $QRM(4)$ by applying Hadamard transversally to all physical qubits, take measurements and correct, the code rewiring approach appears to require fewer gates. However, this is not the full picture when constructing a fault-tolerant implementation. As mentioned in Section 4, each elementary code rewiring requires the use of non-CSS codes at some point, even if the code upon which we are operating is a CSS code. This means that the Steane method of error correction cannot be used for all measurements [22] - indeed it can only be used for at most one measurement as in the other two, either the code is not CSS or the observable is a mixture of X and Z operators. Hence at least two measurement must be made at least three times and a total of at least 7 measurements is needed.

The method in [2] takes advantage of the fact that the $QRM(4)$ code is CSS and employs the Steane approach to fault-tolerant error correction. This approach allows the ancilla qubits to be measured in one round of measurements so that all of the stabilizer measurements can be determined classically. This means that the method in [2] needs 15 measurements to implement the Hadamard gate.

However, the measurement in [2] can be used to correct all X-type errors while the process proposed here does not correct any errors. In both procedures a further round of Steane measurements is needed to correct for the Z errors.

In terms of overhead, the difference in the two approaches reduces to applying a transversal Hadamard gate in [2], or making seven (lower weight) measurements as described here. A fuller investigation of the accuracy thresholds would be needed to determine the more efficient approach. However the code rewiring approach may therefore be competitive in certain architectures.

In this paper we have considered only qubit stabilizer codes. However, these ideas might be extended to qudit codes, given an appropriate extension of the concept of stabilizers (as described for instance in [23, 24]).

Quantum $[[n, k, d]]$ LDPC codes are considered good [17] if the parameters scale as $k = \Theta(n)$ and $d = \Theta(n)$ while the weight of each stabiliser is $O(1)$. By using half of the encoded logical qubits as gauge qubits, one can view these LDPC codes as subsystems while retaining an appropriate scaling of the distance and number of logical qubits. The code rewiring approach would therefore allow implementation of logical Clifford operators fault-tolerantly. There are two issues with this approach. One is that it does not provide a method to implement a non-Clifford operator. The other is that there is no immediate guarantee that the intermediate codes will have low-weight stabilizers. Nevertheless, further research in this area may identify codes for one can achieve stabilisers of low weight in the intermediate codes, which could raise the possibility of using code rewiring with quantum LDPC codes.

References

[1] H. Bombin. "Clifford Gates by Code Deformation". New Journal of Physics **13**, 043005 (2011).

[2] Adam Paetznick and Ben W. Reichardt. "Universal Fault-Tolerant Quantum Computation with Only Transversal Gates and Error Correction". Physical Review Letters **111**, 090505 (2013).

[3] Kristina R. Colladay and Erich J. Mueller. "Rewiring stabilizer codes". New Journal of Physics **20**, 083030 (2018).

[4] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". SIAM Journal on Computing **26**, 1484–1509 (1997).

[5] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. Pages 212–219. STOC '96New York, NY, USA (1996). Association for Computing Machinery.

[6] Laird Egan, Dripto M. Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R. Brown, Marko Cetina, and Christopher Monroe. "Fault-tolerant control of an error-corrected qubit". Nature **598**, 281–286 (2021).

[7] Lukas Postler, Sascha Heuβen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth, Christian D. Marciniak, Roman Stricker, Martin Ringbauer, Rainer Blatt, Philipp Schindler, Markus Müller, and Thomas Monz. "Demonstration of fault-tolerant universal quantum gate operations". Nature **605**, 675–680 (2022).

[8] Rajeev Acharya, Igor Aleiner, Richard Allen et al. "Suppressing quantum errors by scaling a surface code logical qubit". Nature **614**, 676–681 (2023).

[9] Panos Aliferis, Daniel Gottesman, and John Preskill. "Quantum accuracy threshold for concatenated distance-3 codes" (2005). arxiv:quant-ph/0504218.

[10] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. "Roads towards fault-tolerant universal quantum computation". Nature **549**, 172–179 (2017).

[11] Bryan Eastin and E. Knill. "Restrictions on transversal encoded quantum gate sets.". Physical review letters (2009).

[12] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. "Fault-tolerant conversion between the Steane and Reed-Muller quantum codes". Physical Review Letters **113**, 080501 (2014).

[13] Ashley M. Stephens, Zachary W. E. Evans, Simon J. Devitt, and Lloyd C. L. Hollenberg. "Asymmetric quantum error correction via code conversion". Physical Review A **77**, 062335 (2008).

[14] H. Bombin and M. A. Martin-Delgado. "Quantum Measurements and Gates by Code Deformation". Journal of Physics A: Mathematical and Theoretical **42**, 095302 (2009).

[15] Christophe Vuillot, Lingling Lao, Ben Criger, Carmen García Almudéver, Koen Bertels, and Barbara M. Terhal. "Code deformation and lattice surgery are gauge fixing". New Journal of Physics **21**, 033028 (2019).

[16] Stergios Koutsioumpas, Darren Banfield, and Alastair Kay. "The Smallest Code with Transversal T" (2022). arxiv:2210.14066.

[17] Nikolas P. Breuckmann and Jens Niklas Eberhardt. "Quantum Low-Density Parity-Check Codes". PRX Quantum **2**, 040101 (2021).

[18] David Poulin. "Stabilizer Formalism for Operator Quantum Error Correction". Physical Review Letters **95**, 230504 (2005).

[19] Daniel Gottesman. "The Heisenberg Representation of Quantum Computers" (1998). arxiv:quant-ph/9807006.

[20] Daniel Gottesman. "A Theory of Fault-Tolerant Quantum Computation". Physical Review A **57**, 127–137 (1998).

[21] Peter W. Shor. "Fault-tolerant quantum computation" (1997). arxiv:quant-ph/9605011.

[22] A. M. Steane. "Active Stabilization, Quantum Computation, and Quantum State Synthesis". Physical Review Letters **78**, 2252–2255 (1997).

[23] Daniel Gottesman. "Fault-Tolerant Quantum Computation with Higher-Dimensional Systems". Chaos, Solitons & Fractals **10**, 1749–1758 (1999). arxiv:quant-ph/9802007.

[24] Earl T. Campbell, Hussain Anwar, and Dan E. Browne. "Magic-State Distillation in All Prime Dimensions Using Quantum Reed-Muller Codes". Physical Review X **2**, 041021 (2012).