T-Count Optimization and Reed–Muller Codes

Matthew Amy and Michele Mosca

Abstract— **In this paper, we study the close relationship between Reed–Muller codes and single-qubit phase gates from the perspective of T-count optimization. We prove that minimizing the number of T gates in an n-qubit quantum circuit over CNOT and T, together with the Clifford group powers of T, corresponds to finding a minimum distance decoding of a length $2^n - 1$ binary vector in the order $n - 4$ punctured Reed–Muller code. Moreover, we show that the problems are polynomially equivalent in the length of the code. As a consequence, we derive an algorithm for the optimization of T-count in quantum circuits based on Reed–Muller decoders, along with a new upper bound of $O(n^2)$ on the number of T gates required to implement an n-qubit unitary over CNOT and T gates. We further generalize this result to show that minimizing small angle rotations corresponds to decoding lower order binary Reed–Muller codes. In particular, we show that minimizing the number of $R_Z(2\pi/m)$ gates for any integer m is equivalent to minimum distance decoding in $\mathcal{RM}(n - k - 1, n)^*$, where k is the highest power of 2 dividing m.**

Index Terms— **Quantum circuits, Reed-Muller codes, minimum distance decoding, circuit optimization.**

I. Introduction

THE synthesis and optimization of quantum circuits has generated a great deal of interest in recent years. As qubit technologies become more stable and experimentalists increase the size of their systems, actually running algorithms on these machines becomes a practical concern. Moreover, we want to know how to *efficiently* run these algorithms on the given systems, or conversely how big and stable of a system we need to run a particular algorithm. Given the prevalence of the circuit model within quantum computing, quantum circuit optimization is an important tool in answering these questions.

Due to the great affect of noise on quantum computations, much research has shifted its focus from optimizing physical circuits to logical ones with respect to a fault-tolerance schemes meant to mitigate the errors due to this noise. These schemes usually have striking differences from physical gates in terms of resource costs. In particular, most of the common schemes implement Clifford group gates *transversally* – that is, by performing one physical gate on each physical qubit or group of qubits. This allows the logical operation to be performed precisely and with time proportional to the physical gate time. The additional operation needed to make a universal gate set is then typically implemented probabilistically with state distillation and gate teleportation, a less accurate procedure which requires both additional time and space compared to a single physical gate. The two qubit controlled-NOT (CNOT) gate, as an element of the Clifford group, is hence a relatively cheap operation in this paradigm, compared to the $T = \mathrm{diag}(1, e^{i\frac{\pi}{4}})$ gate which is commonly chosen as the non-Clifford gate. This is a reversal of the computational costs inherent in most physical implementations, where entangling gates are typically more difficult to implement than single qubit rotations, and hence requires different circuit optimizations. While alternative fault-tolerance methods such as Paetznick and Reichardt's completely transversal Clifford+T scheme [1] and anyonic quantum computing [2] are gaining in popularity, minimization of the number of T gates – called the T-count – in quantum circuits remains an important and widely studied goal.

We build on previous work by Amy, Maslov and Mosca on the reduction of T gates in quantum circuits. In [3] it was shown that unitaries implementable over CNOT and T gates may be described as a (linear) permutation together with a phase rotation that is an 8th root of unity given by a pseudo-Boolean function of the input bits in the computational basis. This function, called the circuit's *phase polynomial*, was shown to be expressible as a weighted sum of linear Boolean functions, each function corresponding to the application of a T gate to a power given by its weight. This idea was later used in [4] to optimize both T-count and T-depth – the minimal number of stages of parallel T-gates in a circuit – by computing a circuit's phase polynomial, simplifying it, then synthesizing a new circuit from the polynomial with maximally parallelized T gates. While their benchmarks showed significant reduction of T gates, it was noted that this approach was not optimal, as it was shown that there exist distinct phase polynomials that give rise to the same unitary. In particular, it was observed that for all $x_1, x_2, x_3, x_4 \in \mathbb{Z}_2$,

$$e^{i\frac{\pi}{4}\sum_{f\in\mathbb{Z}_2^4\to\mathbb{Z}_2} f(x_1,x_2,x_3,x_4)} = 1 = e^0,$$

Manuscript received March 5, 2018; accepted March 5, 2019. Date of publication March 20, 2019; date of current version July 12, 2019. M. Amy was supported by the Natural Sciences and Engineering Research Council of Canada. M. Mosca was supported in part by Canada's NSERC, CIFAR, and CFI.

M. Amy is with the Institute for Quantum Computing, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and also with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: meamy@uwaterloo.ca).

M. Mosca is with the Institute for Quantum Computing, University of Waterloo, Waterloo, ON N2L 3G1, Canada, also with the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON N2L 3G1, Canada, also with the Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada, and also with the Canadian Institute for Advanced Research, Toronto, ON M5G 1M1, Canada

Communicated by M.-H. Hsieh, Associate Editor for Quantum Information Theory.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2019.2906374

where $\mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ is the space of all n-bit linear Boolean functions. It was left as an open question as to whether there exist other such identities, and whether such identities can be used to further reduce instances of T gates.

In this paper we fully characterize the set of identities between phase polynomials on n qubits. In doing so, we find that the set of identities on n qubits that are useful for reducing a circuit's T-count correspond exactly to the code-words of the length $2^n - 1$ punctured Reed-Muller code of order $n - 4$. This allows us to derive a new T-count optimization algorithm based on Reed-Muller decoding which is optimal for CNOT and T gate circuits when a minimum distance decoder is used. We implemented this optimization algorithm as a module in the quantum circuit optimizer T-par [5] and tested it on general Clifford+T circuits with two different Reed-Muller decoders. The results show modest reductions in T-count while still remaining tractable for circuits of non-trivial size, further confirming the efficacy of the (polynomial-time) T-par algorithm [4] in terms of T-count optimization. Our result further provides a new quadratic upper bound on the number of T gates required to implement a circuit over $\{\text{CNOT}, T\}$, along with evidence towards the intractability of exact T-count minimization via a polynomial-time equivalence to the minimum distance decoding problem for the punctured Reed-Muller code.

Our proof naturally generalizes to the case when the T gate is replaced with a Z rotation by any angle of the form $2\pi/2^k$. These gate sets are closely related to the Clifford-cyclotomic gate sets studied in [6], and are widely used in quantum algorithms including Shor's algorithm [7]. We show that minimizing the number of $2\pi/2^k$ rotation gates for each value of k corresponds to decoding punctured Reed-Muller codes of order $n - k - 1$, opening up the possibility of optimizing such circuits at the high level before decomposing them into a lower level gate set such as Clifford+T. We further show that these are the *only* non-trivial identities between phase polynomials over arbitrary angles – in particular, minimizing rotation gates of any finite order m reduces to the case of order $2\pi/2^k$, where k is the largest integer such that $2^k \mid m$.

A. Related Work

Much work has gone into T-count and depth reduction in recent years. Amy *et al.* [3] identified the T-count and T-depth as important quantities in the efficiency of a logical quantum circuit, and gave new implementations of 2–4 bit quantum operations reducing T-count and depth from the previously best known. Their search-based algorithm was later extended by Gosset *et al.* [8] to directly optimize T-depth, leading to proofs of T-depth minimality for various 2–4 bit circuits. Selinger [9] showed that the Toffoli gate, as well as a general class of Clifford+T circuits, can be parallelized to T-depth 1 with sufficiently many ancillas. Constructions for adding controls to quantum gates were also given which lowered the T-count and depth compared to best known practices using Toffoli gates. Amy, Maslov and Mosca later used similar ideas to create an automated, polynomial-time tool for reducing and parallelizing T gates called T-par, which uses matroid partitioning to parallelize the T gates. More recently, Abdessaied *et al.* [10] studied the effect of Hadamard gates on T-count and depth reductions, developing a tool that reduces Hadamard gates in quantum circuits leading to further T gate optimizations. Maslov [11] examined Toffoli gate implementations up to relative phase and used them to develop new designs for multiple control Toffolis using fewer ancillas, CNOT, and in some cases T gates, than standard designs.

A great deal of work optimizing T-count and depth in single qubit circuits has also been done recently, with series of works on exact [12] and approximate [13]–[15] minimal synthesis, as well as repeat-until-success circuits [16], [17]. While we instead focus on *multi-qubit* circuit optimization, the single- and multi-qubit approaches are complementary as circuits may be first optimized at the level of abstract, small angle rotations before optimally decomposing such gates into sequences of T and Clifford group gates.

The relationship between Reed-Muller codes and T gates has previously been studied from the perspective of fault-tolerance, with applications to the construction of quantum error correcting codes with transversal roots of Z [18]–[20] or otherwise implementing such gates with magic state distillation [21]–[23]. Our work differs from the work done in the fault-tolerance community in that we are interested in the optimization of quantum circuits, rather than implementing phase gates fault-tolerantly – hence we establish *completeness* results in addition to the *existence* results found in fault-tolerance research.

B. Overview

The rest of the paper is organized as follows. Section II gives definitions and notation that will be used throughout the paper. Section III defines the linear phase operators, details their representation as weighted sums of linear Boolean functions and synthesis. Section IV defines an additive subgroup of $\mathbb{Z}_8^{2^n-1}$ whose cosets correspond to the unique linear phase operators, then characterizes its binary residue as a Reed-Muller code and gives applications. Section V generalizes the result to circuits over CNOT gates and phase rotations with angles of the form $2\pi/m$, and Section VI details the experimental evaluation of our technique.

II. Preliminaries

We assume some knowledge of quantum computing and coding theory, but provide the basic necessary definitions from both. For a complete introduction to quantum computing, the reader is referred to Nielsen and Chuang [24], and for background on coding theory see MacWilliams and Sloane [25].

A. Quantum Circuits

We work in the circuit model of quantum computation [24]. The state of an n-qubit quantum system is modelled as a unit vector in a dimension 2^n complex vector space. As is standard we denote the 2^n basis vectors of the *computational* basis by $|\mathbf{x}\rangle$ for bit strings $\mathbf{x} = x_1x_2\cdots x_n \in \mathbb{Z}_2^n$ – these are called the *classical* states. We denote binary vectors by boldface letters and use them interchangeably as bit strings. A general quantum state may be written as a *superposition* of classical states

$$|\psi\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_2^n} \alpha_{\mathbf{x}}|\mathbf{x}\rangle,$$

for complex $\alpha_{\mathbf{x}}$ and having unit norm.

Quantum gates, analogous to classical gates, correspond to unitary matrices on some 2^m dimensional complex vector space. An m-qubit gate may be lifted to a gate on some m-qubit subset of an n-qubit system by taking its tensor product with the identity matrix on the unaffected qubits. By a quantum circuit over a particular set of gates we mean a sequential list of gates taken from the set, each with a list of qubits the gate is to be applied to. Such a circuit *implements* a unitary operator on n qubits, defined as the (sequential) product of each gate appropriately lifted to n qubits. In this way, two distinct circuits may implement the same unitary matrix – we call such circuits *equivalent*.

In this paper we will primarily be interested in two gates: the controlled-NOT (CNOT : $|x\rangle|y\rangle \mapsto |x\rangle|x \oplus y\rangle$ where $\oplus$ denotes addition in $\mathbb{Z}_2$) and the T-gate ($T : |x\rangle \mapsto e^{i\frac{\pi}{4}x}|x\rangle$). These two gates, together with $S := T^2$ and $Z := T^4$ gates, comprise what we refer to for brevity as the $\{\text{CNOT}, T\}$ gate set. We include the S and Z gates in this set to distinguish them from sequences of T gates which are generally much more expensive to implement in most fault-tolerance schemes. Given any power $k \in \mathbb{Z}_8$ of the T gate, we define a minimal T-gate expansion by

$$T^k := Z^{k_2}S^{k_1}T^{k_0}$$

where $k_2k_1k_0$ is the binary expansion of k. Note that $T^8 = I$ so $T^k = T^{k \mod 8}$ for all integers k.

The problem of optimizing quantum circuits is to find, given a circuit, an equivalent circuit minimizing some cost function. In cases where the cost function assigns some non-zero cost to a particular gate U while all other gates are free we refer to the resulting optimization problem as the *U-gate minimization* problem. In this work we primarily consider the problem of T-gate minimization over the $\{\text{CNOT}, T\}$ gate set. It should be noted that, while the $\{\text{CNOT}, T\}$ gate set is not universal in the sense that not every n-qubit unitary can be implemented to arbitrary accuracy with a polylogarithmic number of CNOT and T gates, the addition of the Hadamard gate ($H : |x\rangle \mapsto \frac{1}{\sqrt{2}}\sum_{x'\in\mathbb{Z}_2}(-1)^{xx'}|x'\rangle$) gives a universal set known as Clifford+T.

B. Coding Theory

A length n *binary linear code* is a subspace C of $\mathbb{Z}_2^n$, where $\mathbb{Z}_2$ is the unique 2-element field $(\{0, 1\}, \oplus, \cdot)$ with addition $(\oplus)$ and multiplication $(\cdot)$ modulo 2. The elements of C are called the *codewords* of C. Note that $\mathbb{Z}_2$ is the set of Boolean values with addition corresponding to exclusive-OR and multiplication corresponding to AND. Addition and multiplication are extended to vectors component-wise – that is, $\mathbf{xy}$ is the component-wise multiple of vectors $\mathbf{x}$ and $\mathbf{y}$, as opposed to matrix multiplication.

We denote binary vectors by boldface letters e.g., $\mathbf{x} = x_1x_2\cdots x_n \in \mathbb{Z}_2^n$, and use them interchangeably as bit strings. In particular, we denote the n-qubit computational basis vectors by $|\mathbf{x}\rangle$ where $\mathbf{x}$ is a binary vector/bit string. The *(Hamming) weight* of a binary vector, denoted $|\mathbf{x}|, \mathbf{x} \in \mathbb{Z}_2^n$, is defined as the number of non-zero entries it contains, and the *(Hamming) distance* between two binary vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_2^n$ is the weight of their sum:

$$\delta(\mathbf{x}, \mathbf{y}) := |\mathbf{x} \oplus \mathbf{y}|.$$

Given a received vector $\mathbf{x} \oplus \boldsymbol{e}$ where $\mathbf{x} \in C$ and $\boldsymbol{e} \in \mathbb{Z}_2^n$ is some error vector, we wish to find $\mathbf{x}$ – this process is known as *decoding*. In this work, we are only concerned with *minimum distance decoding*, as it relates directly to T-count optimization.

Definition 1. *Given a binary linear code C and vector $\mathbf{x} \in \mathbb{Z}_2^n$, a* minimum distance decoding *of $\mathbf{x}$ in C is a codeword $\mathbf{y} \in C$ such that for all $\mathbf{z} \in C$, $\delta(\mathbf{x}, \mathbf{y}) \leq \delta(\mathbf{x}, \mathbf{z})$.*

The problem of finding a minimum distance decoding is closely related to the more general *closest vector problem* over a lattice, and in fact coincides with the closest vector problem over the lattice C with the Hamming weight as the norm. Minimum distance decoding is commonly studied as it reasonably approximates maximum likelihood decoding when bit flip errors are independent of one another.

We give one more definition from coding theory which will be relevant to our work: the maximum distance of any vector from a codeword, called the *covering radius*.

Definition 2. *The* covering radius *of a length n binary code C is*

$$\rho(C) = \max_{\mathbf{x}\in\mathbb{Z}_2^n} \min_{\mathbf{y}\in C} \delta(\mathbf{x}, \mathbf{y}).$$

C. Reed-Muller Codes

Many different presentations of the binary Reed-Muller codes [26], [27] are known; we use a presentation based on multivariate polynomials as it will provide a convenient setting for our work. For more details the reader is referred to [25].

Let $\mathbb{Z}_2[X_1, X_2, \ldots, X_n]$ be the ring of polynomials in n variables over $\mathbb{Z}_2$. We use the symbols $X_1, X_2, \ldots, X_n$ to denote formal variables so as to differentiate them from elements of binary vectors. Given $f \in \mathbb{Z}_2[X_1, X_2, \ldots, X_n]$ we define the *evaluation vector* of f, when viewed as an n-ary function, to be the length $2^n - 1$ vector consisting of the evaluation of f at all non-zero inputs ordered lexicographically – i.e.

$$(f(10\cdots0), f(01\cdots0), \ldots, f(11\cdots1)).$$

We denote the evaluation vector of a polynomial function f by $\mathbf{f}$. Since $X^2 = X$ for all $X \in \mathbb{Z}_2$, we work in the quotient ring $f \in \mathbb{Z}_2[X_1, X_2, \ldots, X_n]/\langle X_1^2 - X_1, \ldots, X_n^2 - X\rangle$ and assume polynomials are in reduced form with exponents 0 or 1. Identifying the variable X_i with the Boolean function $f(X_1, X_2, \ldots, X_n) = X_i$, we denote the evaluation vector of X_i by $\mathbf{X}_i$. It can be easily verified that for any Boolean polynomial $f = \bigoplus_{\mathbf{y}\in\mathbb{Z}_2^n} X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}$, the evaluation vector of f

TABLE I

EVALUATION VECTORS FOR MONOMIALS OVER 3 BOOLEAN VARIABLES

	100	010	110	001	101	011	111
1	1	1	1	1	1	1	1
X_1	1	0	1	0	1	0	1
X_2	0	1	1	0	0	1	1
X_1X_2	0	0	1	0	0	0	1
X_3	0	0	0	1	1	1	1
X_1X_3	0	0	0	0	1	0	1
X_2X_3	0	0	0	0	0	1	1
$X_1X_2X_3$	0	0	0	0	0	0	1

is equal to $\bigoplus_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ – again, exponentiation of a Boolean vector is defined as component-wise exponentiation.

We define the *total degree* of a monomial $X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}$ to be the sum of its exponents:

$$\deg(X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}) = \sum_{i=1}^{n} y_i = |\mathbf{y}|.$$

The degree of a polynomial function $f \in \mathbb{Z}_2[X_1, X_2, \cdots X_n]$, denoted $\deg(f)$, is defined as the maximum total degree of each monomial. Table I illustrates the evaluation vectors of the 2^3 monomials on 3 variables. Note that the set of non-constant monomial evaluation vectors are linearly independent and form a basis for the space $\mathbb{Z}_2^{2^n-1}$.

Definition 3. *The* punctured Reed-Muller code *of order r and length $2^n - 1$, denoted $\mathcal{RM}(r, n)^*$, is the set of evaluation vectors for polynomials $f \in \mathbb{Z}_2[X_1, X_2, \ldots, X_n]$ of degree at most r.*

The non-punctured, length 2^n Reed-Muller code or order r is defined in a similar fashion, using evaluation vectors consisting of all 2^n distinct evaluations for a given polynomial function instead.

III. LINEAR PHASE OPERATORS

In this section we introduce linear phase operators as the subset of unitaries implementable by $\{\text{CNOT}, T\}$ which require T gates. We review their representation as pseudo-Boolean functions and define the canonical T-count for a particular polynomial. Finally we show that a minimal T-count implementation of a linear phase operator corresponds to a minimal weight vector of a vector space coset.

We define $\mathcal{P}_8(n)$ to be the set of diagonal $2^n \times 2^n$ unitaries implementable over $\{\text{CNOT}, T\}$ – we restrict our attention to this subset as any circuit over $\{\text{CNOT}, T\}$ may be decomposed into a diagonal unitary followed by a permutation implementable using only CNOT gates. Amy *et al.* [3, Lemma 2] showed that each such unitary $U \in \mathcal{P}_8(n)$ has the effect of applying a pseudo-Boolean function P to a computational basis state $|\mathbf{x}\rangle$, viewed as a vector $\mathbf{x} = x_1x_2\cdots x_n \in \mathbb{Z}_2^n$, and kicking the result into the phase:

$$U : |\mathbf{x}\rangle \mapsto e^{i\frac{\pi}{4}P(\mathbf{x})}|\mathbf{x}\rangle.$$

Moreover, it was shown that the *phase polynomial* $P : \mathbb{Z}_2^n \to \mathbb{Z}_8$ necessarily has a presentation as a weighted sum of (non-zero) linear Boolean functions:

$$P(\mathbf{x}) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{0}\}} a_{\mathbf{y}}(y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n),$$

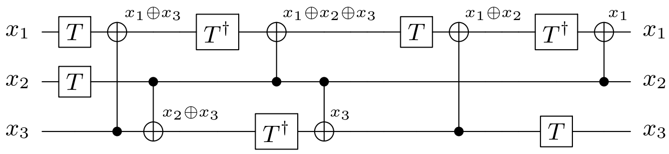
where the coefficients $a_{\mathbf{y}}$ are integers modulo 8. We call the tuple $\mathbf{a} = (a_1, a_2, \ldots, a_{2^n-1}) \in \mathbb{Z}_8^{2^n-1}$ an *implementation* of P, and conversely denote the phase polynomial defined by an element $\mathbf{a}$ of $\mathbb{Z}_8^{2^n-1}$ by $P_{\mathbf{a}}$. As the function P involves both $\mathbb{Z}_2$ and $\mathbb{Z}_8$ arithmetic, we implicitly use the natural inclusion of $\mathbb{Z}_2$ in $\mathbb{Z}_8$ to lift the binary valued result of $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$ into an integer.

We call unitaries in $\mathcal{P}_8(n)$ $\pi/4$ *linear phase operators*, as they may be expressed as a sequence of $\pi/4$ phase rotations conditioned on linear Boolean functions of the input basis state. We drop the $\pi/4$ until Section V when we generalize the result to $2\pi/2^k$ linear phase operators. Given a particular phase polynomial P, we denote the linear phase operator with phase polynomial P by U_P.

Example 1. *The doubly-controlled Z gate is a $\pi/4$ linear phase operator with phase function $P(x_1, x_2, x_3) = 4x_1x_2x_3$. Using the identity $2 \cdot xy = x + y + 7(x \oplus y) \mod 8$ [9], the phase function may be given as the following weighted sum of linear Boolean functions:*

$$P(x_1, x_2, x_3) = x_1 + x_2 + 7(x_1 \oplus x_2) + x_3 + 7(x_1 \oplus x_3) \\ + 7(x_2 \oplus x_3) + (x_1 \oplus x_2 \oplus x_3).$$

Writing the coefficients above as a 7-tuple over $\mathbb{Z}_8$ we get $(1, 1, 7, 1, 7, 7, 1)$. *Note that this implementation corresponds to the following circuit, taken from [4]. The state of each qubit after an update is shown to illustrate the relation between the state of a qubit as a Boolean function of the inputs and the application of phase gates.*



Amy*et al.* [4] showed that a linear phase operator U_P can be synthesized over $\{\text{CNOT}, T\}$ given an implementation $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ of P in time polynomial in the number of non-zero entries of $\mathbf{a}$ – moreover, this number is linear in the size of the circuit. Their procedure applies each (non-trivial) phase shift $e^{i\frac{\pi}{4}a_{\mathbf{y}}(y_1x_1\oplus y_2x_2\oplus\cdots\oplus y_nx_n)}$ by first computing the linear sum $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$, then applying $T^{a_{\mathbf{y}}}$ and uncomputing $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$. Recall that

$$T^k := Z^{k_2}P^{k_1}T^{k_0}$$

where $k_2k_1k_0$ is the binary expansion of k. Since each $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$ is a linear function of the basis state $x_1x_2\cdots x_n$, each value $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$ may be computed solely with CNOT gates, giving a total T-count equal to the number of odd elements of $\mathbf{a}$ – we call this the T-count of an implementation. While in this work we are only concerned with the T-count of the synthesized circuit, T-depth can be minimized while keeping T-count the same by parallelizing this process through matroid partitioning [4].

The authors used this synthesis algorithm to optimize T-count in $\{\text{CNOT}, T\}$ circuits by first computing a set of coefficients for the associated linear phase operator U_P from the circuit in polynomial time, then synthesizing an equivalent circuit. The remaining linear permutation is also computed and synthesized separately in polynomial time. This procedure has the crucial property that the element $\mathbf{a}$ of $\mathbb{Z}_8^{2^n-1}$ computed has T-count at most the T-count of the original circuit – often much lower due to coefficients in the phase polynomial adding and reducing modulo 8 – hence the resulting circuit has equal or lesser T-count. In particular, we have the following proposition, which relates the T-count of a {CNOT, T} circuit to the T-count of an implementation of the associated phase operator.

Proposition 1. *Let U_P be a linear phase operator in $\mathcal{P}_8(n)$. There exists a circuit over* {CNOT, T} *implementing U_P with T-count k if and only if there exists $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ such that $P(\mathbf{x}) = P_{\mathbf{a}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{Z}_2^n$, and $\mathbf{a}$ has at most k odd entries.*

IV. Decoding-Based T-Count Optimization

While effective at reducing T-count, it was noted that the procedure in [4] does not always find the minimal T-count, as the phase polynomial P in question may have many different representations as a weighted sum of linear Boolean functions. For instance,

$$4 \cdot x_1 + 4 \cdot x_2 + 4 \cdot (x_1 \oplus x_2) = 0 \mod 8$$

for all values of $x_1, x_2 \in \mathbb{Z}_2$, so $P(x_1, x_2) = 4 \cdot x_1 + 4 \cdot x_2 + 4 \cdot (x_1 \oplus x_2)$ is an alternative presentation of the zero-everywhere ($\pi/4$) phase polynomial. This implies that further T-count optimization may be possible by first finding an implementation of the target phase polynomial with a minimal number of odd coefficients, then synthesizing a circuit. By Proposition 1, this in fact gives a minimal T-count circuit. In this section we reduce this problem to a minimum-distance decoding problem and give a T-count optimization algorithm based on this decoding.

Given an element $\mathbf{a}$ of $\mathbb{Z}_8^{2^n-1}$, let $[\mathbf{a}]$ be the equivalence class of implementations of $P_{\mathbf{a}}$ – i.e., $\mathbf{b} \in [\mathbf{a}]$ if and only if $P_{\mathbf{a}}(\mathbf{x}) = P_{\mathbf{b}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{Z}_2^n$ (hence $U_{P_{\mathbf{a}}} = U_{P_{\mathbf{b}}}$). We define $\mathcal{C}_n$ to be the subset of $\mathbb{Z}_8^{2^n-1}$ giving the zero-everywhere phase polynomial. Note that for any $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_8^{2^n-1}$ and $\mathbf{x} \in \mathbb{Z}_2^n$,

$$P_{\mathbf{a}}(\mathbf{x}) + P_{\mathbf{b}}(\mathbf{x}) = P_{\mathbf{a}+\mathbf{b}}(\mathbf{x}),$$

so $\mathcal{C}_n$ is in fact a subgroup of $\mathbb{Z}_8^{2^n-1}$ and moreover, $[\mathbf{a}] = \mathbf{a} + \mathcal{C}_n$. As a result we see that the problem of finding an implementation of $P_{\mathbf{a}}$ minimizing T count is equivalent to finding an element $\mathbf{c} \in \mathcal{C}_n$ minimizing the number of odd entries in $\mathbf{a} + \mathbf{c}$.

In order to find such elements of the coset $\mathbf{a} + \mathcal{C}_n$, we first need a characterization of the subgroup $\mathcal{C}_n$. The following lemma gives an explicit set of generators for $\mathcal{C}_n$ as scaled monomial evaluation vectors of particular degrees, giving a type of generalized Reed-Muller code. As the proof is quite technical we give it in Appendix

Lemma 1. *$\mathcal{C}_n$ is generated by*

$$\{2^i \mathbf{X}_1^{y_1} \mathbf{X}_2^{y_2} \cdots \mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n, |\mathbf{y}| - i \leq n - 4\}.$$

Lemma 1 gives an exact definition of $\mathcal{C}_n$ as

$$\mathcal{C}_n = \left\langle 2^i \mathbf{X}_1^{y_1} \mathbf{X}_2^{y_2} \cdots \mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n, |\mathbf{y}| - i \leq n - 4 \right\rangle.$$

With the above characterization, optimization can be performed directly over $\mathcal{C}_n$, though the particular metric of T-count optimality makes such optimization unnatural. As the number of odd entries in an element of $\mathbb{Z}_8^{2^n-1}$ does not define a norm, there does not appear to be a natural reduction to lattice problems. Likewise, the number of odd coefficients does not make a natural distance metric for (ring) linear codes.

We can instead reduce the optimization problem to a decoding problem over a *binary* code where minimum-distance decoding corresponds exactly to T-count optimization. Defining $\mathrm{Res}_2 : \mathbb{Z} \to \mathbb{Z}_2$ as the function taking the binary residue of an integer and extending this component-wise to tuples, we see that the number of odd entries in $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ is equal to the weight of the binary residue vector, i.e. $|\mathrm{Res}_2(\mathbf{a})|$. We can further see that

$$|\mathrm{Res}_2(\mathbf{a} + \mathbf{c})| = \delta(\mathrm{Res}_2(\mathbf{a}), \mathrm{Res}_2(\mathbf{c})),$$

that, is the T-count of $P_{\mathbf{a}+\mathbf{c}}$ is the Hamming distance from $\mathrm{Res}_2(\mathbf{a})$ to $\mathrm{Res}_2(\mathbf{c})$.

Hence, optimizing the number of odd entries in $\mathbf{a}+\mathbf{c}$ over all $\mathbf{c} \in \mathcal{C}_n$ is exactly the problem of minimum distance decoding $\mathrm{Res}_2(\mathbf{a})$ over $\mathrm{Res}_2(\mathcal{C}_n)$, the set of binary residue vectors of $\mathcal{C}_n$. We further note that $\mathrm{Res}_2(\mathcal{C}_n)$ is a binary linear code, since $\mathrm{Res}_2(\mathbf{a}) \oplus \mathrm{Res}_2(\mathbf{b}) = \mathrm{Res}_2(\mathbf{a}+\mathbf{b}) \in \mathrm{Res}_2(\mathcal{C}_n)$ for any $\mathbf{a}, \mathbf{b} \in \mathcal{C}_n$, and as a direct consequence of Lemma 1 this code is exactly the $(n-4)$th order, length $2^n - 1$ punctured Reed-Muller code.

Theorem 1. $\mathrm{Res}_2(\mathcal{C}_n) = \mathcal{RM}(n-4, n)^*$

The remainder of this section discusses some consequences of Theorem 1.

A. Upper Bounds

As a consequence of Proposition 1 and Theorem 1, the covering radius of $\mathrm{Res}_2(\mathcal{C}_n) = \mathcal{RM}(n-4, n)^*$ gives a tight upper bound on the number of T gates required to implement a linear phase operator over {CNOT, T}. Here we mean tight in the sense that there exists a linear phase operator which requires a minimum of $\rho(\mathcal{RM}(n-4, n)^*)$ T gates to implement over {CNOT, T}. While to the best of the authors' knowledge no analytic formula has been found for the covering radius of higher-order Reed-Muller codes, some asymptotic upper bounds are known. In particular, Cohen and Litsyn [28] showed that for large n and orders r where $n - r \geq 3$,

$$\rho(\mathcal{RM}(r, n)) \leq \frac{n^{n-r-2}}{(n-r-2)!}.$$

Since the covering radius of $\mathcal{RM}(r, n)^*$ is trivially bounded above by $\rho(\mathcal{RM}(r, n))$, we see that for sufficiently large n, $\rho(\mathcal{RM}(n-4, n)^*) \leq \frac{n^2}{2} - 1$. As a result we obtain a new asymptotic bound on the number of T gates required to implement a circuit over {CNOT, T}.

Theorem 2. *Any linear phase operator $U_p \in \mathcal{P}_8(n)$ can be implemented with $O(n^2)$ T-gates.*

B. T-Count Optimization

While the minimal T-count of a given phase polynomial $P_{\mathbf{a}}$ can be obtained by finding a minimum distance decoding of $\mathrm{Res}_2(\mathbf{a})$ in $\mathcal{RM}(n-4, n)^*$, the decoding itself is not enough to synthesize a minimal T-count circuit. In particular, decoding the binary residue $\mathrm{Res}_2(\mathbf{a})$ of a target tuple $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ over

$\mathcal{RM}(n-4,n)^*$ produces a minimal *residue* $\mathbf{w} = \mathrm{Res}_2(\mathbf{c})$ of a codeword $\mathbf{c}$ in $\mathcal{C}_n$. To actually produce a minimal T-count implementation we need to compute $\mathbf{c} \in \mathcal{C}_n$ from $\mathrm{Res}_2(\mathbf{c})$ and then synthesize $\mathbf{a}+\mathbf{c}$. We can achieve this using the following fact:

Proposition 2. *For all* $\mathbf{y} \in \mathbb{Z}_2^n$ *with* $|\mathbf{y}| \leq n-4$,

$$\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \in \mathcal{C}_n.$$

The above proposition, which follows directly from Lemma 1, together with the fact that the monomials of degree at most $n-4$,

$$\mathcal{B} = \{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n, |\mathbf{y}| \leq n-4\},$$

generate $\mathrm{Res}_2(\mathcal{C}_n) = \mathcal{RM}(n-4,n)^*$ allows us to write a decoded word $\mathbf{w}$ as a Boolean polynomial, then *reinterpret* the sum over $\mathbb{Z}_8$ to give a preimage of $\mathbf{w} = \mathrm{Res}_2(\mathbf{c})$ in $\mathcal{C}_n$. Specifically, if $\mathbf{w} = \mathbf{b}_1 \oplus \mathbf{b}_2 \oplus \cdots \oplus \mathbf{b}_k$ for some $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k \in \mathcal{B}$, then we define $\mathbf{c} = \mathbf{b}_1 + \mathbf{b}_2 + \cdots + \mathbf{b}_k$, which by Proposition 2 is in $\mathcal{C}_n$, and further note that

$$\begin{aligned}\mathrm{Res}_2(\mathbf{c}) &= \mathrm{Res}_2(\mathbf{b}_1 + \mathbf{b}_2 + \cdots + \mathbf{b}_k)\\ &= \mathrm{Res}_2(\mathbf{b}_1) \oplus \mathrm{Res}_2(\mathbf{b}_2) \oplus \cdots \oplus \mathrm{Res}_2(\mathbf{b}_k)\\ &= \mathbf{w}\end{aligned}$$

Using this fact we develop an algorithm for the optimization of T-count based on Reed-Muller decoding.

Algorithm 1 T-Optimize(C)

```
1: Compute coefficients $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ from $C$
2: $\mathbf{w} \leftarrow$RM-DECODE($n-4$, $n$, $\mathrm{Res}_2(\mathbf{a})$)
3: Write $\mathbf{w}$ over basis $\mathcal{B}$: $\mathbf{w} = \mathbf{b}_1 \oplus \mathbf{b}_2 \oplus \cdots \oplus \mathbf{b}_k$
4: $\mathbf{c} \leftarrow \mathbf{b}_1 + \mathbf{b}_2 + \cdots + \mathbf{b}_k$ (mod 8)
5: SYNTHESIZE($\mathbf{a}+\mathbf{c}$)
```

Algorithm 1 summarizes our algorithm for T-count optimization in $\{\mathrm{CNOT}, T\}$ circuits. For simplicity the algorithm assumes the input circuit implements a linear phase operator – for more general $\{\mathrm{CNOT}, T\}$ circuits the extra linear permutation is synthesized and appended to the end. The algorithm works by computing a set of coefficients implementing the linear phase operator U_P computed by the circuit. The vector of residues modulo 2 is then decoded as $\mathbf{w}$ in $\mathcal{RM}(n-4,n)^*$ using the procedure RM-DECODE($n-4$, n, $\mathrm{Res}_2(\mathbf{a})$). A vector $\mathbf{c} \in \mathcal{C}_n$ with binary residue equal to $\mathbf{w}$ is then computed and added to the original set of coefficients, and a circuit is synthesized for the new implementation of P. In particular, the procedure SYNTHESIZE takes a set of coefficients $\mathbf{a}$ and synthesizes a circuit over $\{\mathrm{CNOT}, T\}$ implementing $U_{P_\mathbf{a}}$.

The T-optimize algorithm is parametric in both the decoder and the synthesis procedure, meaning any variable order Reed-Muller decoder may be used to implement RM-DECODE. If a minimum distance decoder is used, Algorithm 1 synthesizes a minimal T-count circuit. Likewise, any synthesis procedure may be used to implement SYNTHESIZE – for instance, the T-depth minimizing T-par algorithm [4] can be used.

Example 2. *Consider the circuit in Figure 1. By iterating through the circuit and updating the qubit states (see, e.g., [4]), we compute the phase polynomial for this operator as*

$$\begin{aligned}P(\mathbf{x}) = \;& 2x_1 + 6x_2 + 6(x_1 \oplus x_2) + x_3 + 7(x_1 \oplus x_3)\\ &+7(x_2 \oplus x_3) + (x_1 \oplus x_2 \oplus x_3) + 3x_4\\ &+7(x_1 \oplus x_4) + 7(x_2 \oplus x_4) + (x_1 \oplus x_2 \oplus x_4).\end{aligned}$$

Writing the coefficients of P *as a* 2^n-1*-tuple* $\mathbf{a}$ *we get*

$$\mathbf{a} = (2,6,6,1,7,7,1,3,7,7,1,0,0,0,0),$$

which has a canonical T*-count of* 8 *– a reduction of* 6 T *gates.*

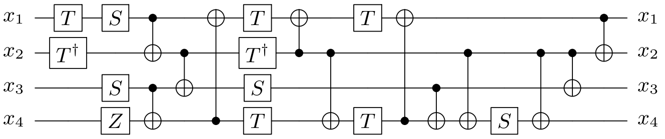
Now we optimize the implementation of P *further by decoding*

$$\mathrm{Res}_2(\mathbf{a}) = (0,0,0,1,1,1,1,1,1,1,1,0,0,0,0)$$

in the code $\mathcal{RM}(0,4)^*$*. As* $\mathcal{RM}(0,4)^*$ *is the set of evaluation vectors for degree 0 binary polynomials, there are exactly two vectors to choose from, corresponding to the zero (zero-everywhere) and constant (one-everywhere) functions. Since the all* 1 *vector achieves the minimum distance of* 7 *from* $\mathrm{Res}_2(\mathbf{a})$*, we choose* $\mathbf{w}$ *to be the all* 1 *vector. By Proposition 2,* $\mathbf{w} = \mathbf{1}$ *(mod 2) is already in the space of zero-everywhere polynomials* $\mathcal{C}_n$*, so steps 3 & 4 are trivial and we set* $\mathbf{c} = 1$ *(mod 8). Finally we synthesize a circuit for the tuple* $\mathbf{a}+\mathbf{c} = (3,7,7,2,0,0,2,4,0,0,2,1,1,1,1)$*, corresponding to the phase polynomial*

$$\begin{aligned}P'(\mathbf{x}) = \;& 3x_1 + 7x_2 + 7(x_1 \oplus x_2) + 2x_3 + 2(x_1 \oplus x_2 \oplus x_3)\\ &+4x_4 + 2(x_1 \oplus x_2 \oplus x_4) + (x_3 \oplus x_4)\\ &+(x_1 \oplus x_3 \oplus x_4) + (x_2 \oplus x_3 \oplus x_4)\\ &+(x_1 \oplus x_2 \oplus x_3 \oplus x_4).\end{aligned}$$

A possible circuit implementing P' *is shown below:*



Note that this decoding reduces the T*-count from* 14 *(or* 8*, as* T*-par type optimizations would obtain) to* 7*. Moreover, the number of* T *gates is equal to the distance from* $\mathrm{Res}_2(\mathbf{a})$ *to the decoded word* $\mathbf{w}$*.*

It is interesting to note that the minimal T-depth of $U_{P'}$ above without additional ancillas is 3, while the minimal ancilla-free T-depth of U_P is 2, even though the number of T gates is reduced. Clearly Algorithm 1, when combined with a T-depth optimal synthesis method such as matroid partitioning, does not necessarily obtain the minimal T-depth for a given circuit. It remains an open question to determine an efficient method of optimizing T-depth over all implementations of a linear phase operator.

C. Complexity

It may be noted that Algorithm 1 gives a polynomial-time (in 2^n) reduction from T-count optimization over $\{\mathrm{CNOT}, T\}$

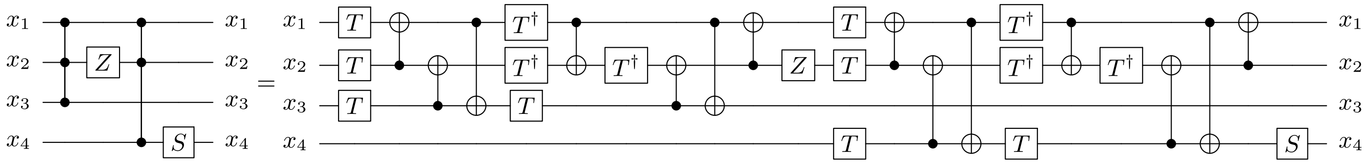


Fig. 1. Implementation of a linear phase operator over Clifford+T.

to minimum-distance decoding in $\mathcal{RM}(n-4,n)^*$. We may likewise reduce the minimum-distance decoding problem for $\mathcal{RM}(n-4,n)^*$ to T-count optimization: given a binary vector $\mathbf{w} \in \mathbb{Z}_2^{2^n-1}$, synthesize $U_{P_\mathbf{w}}$ over $\{\text{CNOT}, T\}$ then optimize the circuit and compute the coefficients $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ for the optimized circuit. As a consequence of Theorem 1, the vector $\mathbf{w} \oplus \text{Res}_2(\mathbf{a})$ is a minimum distance decoding of $\mathbf{w}$. Assuming the optimized circuit does not have exponentially more gates than a canonical circuit,[1] this reduction is also polynomial in the word length 2^n, so we see that the problems are in fact polynomial-time equivalent.

Theorem 3. *The problems of T-count optimization over $\{\text{CNOT}, T\}$ and minimum-distance decoding in $\mathcal{RM}(n-4,n)^*$ are polynomial-time equivalent in the word length 2^n.*

This equivalence lends evidence to the difficulty of T-count optimization, even in the restricted setting of circuits over CNOT and T gates. In particular, as the equivalence is with respect to the number of coefficients in the phase polynomial ($2^n - 1$) any algorithm for exact optimization of T-count over n-qubit $\{\text{CNOT}, T\}$ circuits that is *sub-exponential in n* induces a *polynomial-time* minimum-distance decoding algorithm for $\mathcal{RM}(n-4,n)^*$. This can be further reduced to a *linear-time* algorithm by noting that the unitary $U_{P_\mathbf{w}}$ above can be implemented with $O(2^n)$ gates using one ancilla and the Gray code to cycle through each of the 2^n binary sums of n variables with one CNOT gate each.

In either case it appears very unlikely that an efficient algorithm for minimum-distance decoding the order $n-4$ punctured Reed-Muller code exists. No minimum distance decoding algorithms in time polynomial in 2^n *or* the Hamming weight of the received word are currently known for arbitrary order length 2^n binary Reed-Muller codes. While some particular orders of Reed-Muller codes have efficient decoders, e.g., order 1, it was shown in [29] that minimum-distance decoding for $\mathcal{RM}(n-4,n)^*$ is equivalent to the problem of finding a minimal decomposition of a symmetric 3-tensor into symmetric tryads (rank 1 3-tensors), a known hard problem [29].

V. Rotations of Other Orders

Having shown that minimizing the number of T gates in $\{\text{CNOT}, T\}$ circuits is equivalent to minimum distance decoding in $\mathcal{RM}(n-4,n)^*$, we now turn our attention to circuits with Z-basis rotations of other angles. Specifically, we define the gate $R_Z(2\pi/m)$ for any non-zero integer m by

$$R_Z(2\pi/m) : |x\rangle \mapsto e^{\frac{2\pi i}{m}x}|x\rangle.$$

Such gates arise, e.g., in Shor's algorithm [7] and the Clifford hierarchy [30]. Moreover, researchers have recently developed state distillation techniques for these gates, allowing them to be performed fault tolerantly without approximating them over another gate set [23], [31]. Here we develop methods for the optimization of circuits over CNOT and $R_Z(2\pi/m)$ gates to make use of this higher-level structure of many quantum circuits, whether the rotations are then to be approximated over another gate set or implemented directly.

We define $\mathcal{P}_m(n)$ to be the set of n-qubit $2\pi/m$ linear phase operators – that is, n-qubit diagonal unitary matrices implementable over $\{\text{CNOT}, R_Z(2\pi/m)\}$. As in the case of $\pi/4$ linear phase operators, such an operator applies to each basis vector a phase rotation that is a m-th root of unity determined by a linear combination of linear functions of its bits. In particular, for any $U \in \mathcal{P}_m(n)$, U has the following effect:

$$U : |\mathbf{x}\rangle \mapsto e^{\frac{2\pi i}{m}P(\mathbf{x})}|\mathbf{x}\rangle,$$

$$P(\mathbf{x}) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{0\}} a_\mathbf{y}(y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n)$$

for some coefficients $\mathbf{a} \in \mathbb{Z}_m^{2^n-1}$. As before we call the tuple $\mathbf{a}$ an implementation of P and we denote the set of zero-everywhere phase polynomial implementations $\mathcal{C}_n^m$, defined below as

$$\mathcal{C}_n^m = \{\mathbf{c} \in \mathbb{Z}_m^{2^n-1} | \forall \mathbf{x} \in \mathbb{Z}_2^n, P_\mathbf{c}(\mathbf{x}) = 0 \mod m\}.$$

We first consider the case when $m = 2^k$, which is a natural generalization of Theorem 1. We then examine the case when m is an odd prime power, and finally combine the two results to get a characterization of $\mathcal{P}_m(n)$ for any non-zero integer m.

A. Rotations of Even-Power Order

Recall that Theorem 1 was proven by giving a set of generators for $\mathcal{C}_n = \mathcal{C}_n^{2^3}$. We can use the same methods to assign a set of generators to $\mathcal{C}_n^{2^k}$, and likewise derive a generalization of Theorem 1. In particular, it turns out that $\mathcal{C}_n^{2^k}$ is generated by the set of scaled monomial vectors with degree $n-k-1+i$ and scalar 2^i.

Lemma 2. *$\mathcal{C}_n^{2^k}$ is generated by*

$$\{2^i\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n, |\mathbf{y}| - i \leq n-k-1\}.$$

Again the proof of Lemma 2 is left for Appendix . Further, as in the $\pi/4$ case, Lemma 2 implies the following theorem

[1] The canonical circuit for any linear phase operator uses $O(n2^{n-1})$ gates.

stating that the binary residues of $\mathcal{C}_n^{2^k}$ are exactly the codewords of the order $n-k-1$ punctured Reed-Muller code[2] of length 2^n-1.

Theorem 4. $\mathcal{RM}(n-k-1,n)^* = \mathrm{Res}_2(\mathcal{C}_n^{2^k})$

As a consequence of Theorem 4, Algorithm 1 can be adapted to optimize the number of $R_Z(2\pi/2^k)$ gates in a linear phase circuit. Recall that the canonical circuit for an implementation of a $\pi/4$ linear phase operator was defined by computing $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$ for each nonzero $a_\mathbf{y}$, then applying a sequence of T, P and Z gates to achieve the correct power of $e^{i\frac{\pi}{4}}$. We may define the canonical circuit for an implementation of any $2\pi/2^k$ linear phase operator in the same way: compute $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$ then apply $R_Z(2\pi/2^k)^l$ to achieve the correct power of $e^{i\frac{\pi}{2^k}}$. Under the assumption that $R(2\pi/2^k)$ gates are more expensive than $R(2\pi/2^{k'})$ gates whenever $k > k'$, we define

$$R_Z(2\pi/2^k)^l := R_Z(2\pi)^{l_k} \cdots R_Z(2\pi/2^{l-1})^{l_1} R_Z(2\pi/2^k)^{l_0}$$

where $l_k \cdots l_1 l_0$ is the binary expansion of l. Denoting by $\mathbf{a} \ggg i$ the component-wise quotient of $\mathbf{a}$ divided by 2^i, we find that the number of $R_Z(2\pi/2^l)$ gates in the canonical circuit is $|\mathrm{Res}_2(\mathbf{a} \ggg (k-l))|$ – the number of components $a_\mathbf{y}$ that have a 1 in the $(k-l)$th digit of their binary expansion.

The number of rotation gates of any angle $2\pi/2^l$ for $l \leq k$ may then be reduced by decoding $\mathrm{Res}_2(\mathbf{a} \ggg (k-l))$ in the code $\mathrm{Res}_2(\mathcal{C}_n^l) = \mathcal{RM}(n-l-1,n)$ and adding the decoded tuple back into $\mathbf{a}$ (multiplied by the appropriate power of 2). Such procedures may be a valuable tool for quantum circuits utilizing progressively finer grain Z rotations, such as Shor's algorithm [7], either to be later approximated by Clifford+T gates or to be performed directly using state distillation. One potential issue with this method is reducing the number of $R_Z(2\pi/2^l)$ may increase the number of $R_Z(2\pi/2^{l'})$ gates for any $l' < l$, as seen in Example 2. In most cases smaller angles of rotation are more costly so this is a reasonable trade off, but we leave it as an open question to find a general algorithm for optimizing the total cost of all rotation gates in a $\{\mathrm{CNOT}, R_Z(2\pi/2^k)\}$ circuit.

B. Rotations of Odd Order

A natural question is whether rotation gates of other prime power orders admit similar relationships to known codes. To the contrary, we show that for any odd prime p and integer k, there are no non-trivial phase polynomials that are zero-everywhere mod p^k – equivalently, the set of diagonal operators over $\{\mathrm{CNOT}, R_Z(2\pi/p^k)\}$ is isomorphic to $\mathbb{Z}_{p^k}^{2^n-1}$.

Lemma 3. *For all odd primes p and non-negative integers k, given any non-zero tuple $\mathbf{a} \in \mathbb{Z}_{p^k}^{2^n-1}$, there exists $\mathbf{x} \in \mathbb{Z}_2^n$ such that*

$$P_\mathbf{a}(\mathbf{x}) \neq 0 \mod p^k.$$

To prove Lemma 3, we first introduce the *multilinear representation* of a phase polynomial. In particular, given a tuple $\mathbf{a} \in \mathbb{Z}_{p^k}^{2^n-1}$ the multilinear polynomial function defined by $\mathbf{a}$ is given by

$$Q_\mathbf{a}(\mathbf{x}) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{0}\}} a_\mathbf{y} \cdot x_1^{y_1}x_2^{y_2}\cdots x_n^{y_n}.$$

The result then follows from two facts:

1) there are no non-trivial zero-everywhere multilinear polynomials modulo $\mathbb{Z}_{p^k}$, and
2) for every multilinear polynomial over $\mathbb{Z}_{p^k}$, there exists a unique equivalent phase polynomial over $\mathbb{Z}_{p^k}$.

The first fact follows from the observation that the set of all non-constant monomial evaluation vectors

$$\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} | \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{0}\}\}$$

is linearly independent over any integer ring. In particular, for any $\mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{0}\}$, the vector $\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ contains a leading 1 at the $\mathbf{y}$th index (see e.g., Table I), and hence the set of all such vectors is trivially linearly independent in any integer ring.

Proposition 3. *For all odd primes p and non-negative integers k, given any non-zero tuple $a \in \mathbb{Z}_{p^k}^{2^n-1}$, there exists $\mathbf{x} \in \mathbb{Z}_2^n$ such that*

$$Q_\mathbf{a}(\mathbf{x}) \neq 0 \mod p^k.$$

For the second fact, recall the modular identity

$$2xy = x + y - (x \oplus y) \mod p^k$$

for any $x, y \in \mathbb{Z}_2$. Since 2 is coprime with p^k it has a multiplicative inverse in $\mathbb{Z}_{p^k}$, hence we can rewrite this identity as

$$xy = 2^{-1}x + 2^{-1}y - 2^{-1}(x \oplus y) \mod p^k.$$

The equation above can be used to rewrite a monomial $x_{i_1}x_{i_2}\cdots x_{i_m}$ in the form of a phase polynomial:

$$\begin{aligned}&(x_{i_1}x_{i_2})\cdots x_{i_m}\\ &= (2^{-1}x_{i_1} + 2^{-1}x_{i_2} - 2^{-1}(x_{i_1} \oplus x_{i_2}))\cdots x_{i_n} \mod p^k\\ &= 2^{-1}x_{i_1}\cdots x_{i_n} + 2^{-1}x_{i_2}\cdots x_{i_n} - 2^{-1}(x_{i_1} \oplus x_{i_2})\cdots x_n \mod p^k\end{aligned}$$

where each term in the second line has degree $m-1$ and hence the monomial can be recursively reduced to the form of a phase polynomial. Uniqueness further follows from Proposition 3, as if two distinct multilinear polynomials $Q_\mathbf{a}$ and $Q_\mathbf{b}$ reduced to the same phase polynomial, we would have $Q_{\mathbf{a}-\mathbf{b}}(\mathbf{x}) = 0 \mod p^k$ for all $\mathbf{x} \in \mathbb{Z}_2^n$ but $\mathbf{a} + \mathbf{b} \neq \mathbf{0}$, a contradiction.

Proposition 4. *For any odd prime p and positive integer k, given a tuple $\mathbf{a} \in \mathbb{Z}_{p^k}^{2^n-1}$ there exists some unique $\mathbf{b} \in \mathbb{Z}_{p^k}^{2^n-1}$ such that for all $\mathbf{x} \in \mathbb{Z}_2^n$,*

$$Q_\mathbf{a}(\mathbf{x}) = P_\mathbf{b}(\mathbf{x}) \mod p^k.$$

Note that Proposition 4 *does not* hold for even prime powers p^k, as it requires p^k to be coprime with 2 in order to rewrite a monomial as a weighted sum of parities.

[2] Note that using Definition 3, the Reed-Muller code $\mathcal{RM}(r,n)$ is well defined for $r < 0$. In particular, the code is the trivial code $\{\mathbf{0}\}$, corresponding to the fact that no non-trivial zero phase polynomials exist mod 2^k when $k < n-1$.

Propositions 3 and 4 together imply that there exists an isomorphism between multilinear and phase polynomial representations of pseudo-Boolean functions modulo powers of odd primes, and moreover that there are no non-trivial zero-everywhere multilinear polynomials and hence phase polynomials. We formalize this intuition below.

Proof of Lemma 3. Suppose $\mathbf{a} \in \mathbb{Z}_{p^k}$ is non-zero for some odd prime p and non-negative integer k. By Proposition 4 and the fact that there are the same number of multilinear and phase polynomials over $\mathbb{Z}_{p^k}$, there exists a unique tuple $\mathbf{b} \in \mathbb{Z}_{p^k}^{2^n-1}$ such that $P_{\mathbf{a}}(\mathbf{x}) = Q_{\mathbf{b}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{Z}_2^n$. Now by Proposition 3, there exists $\mathbf{x} \in \mathbb{Z}_2^n$ such that

$$P_{\mathbf{a}}(\mathbf{x}) = Q_{\mathbf{b}}(\mathbf{x}) \neq 0 \mod p^k$$

as required. □

From the perspective of optimizing phase gates, Lemma 3 asserts that each element of $\mathbb{Z}_{p^k}^{2^n-1}$ corresponds to a unique n-qubit unitary implementable over CNOT and $R(2\pi/p^k)$. Given such a circuit, an implementation minimizing the number of the minimal number of $R(2\pi/p^k)$ may then be obtained by first computing the corresponding element of $\mathbb{Z}_{p^k}^{2^n-1}$ and resynthesizing, which can be performed in polynomial time.

C. Rotations of Arbitrary Order

It is worth noting that Lemma 3 above is in a sense the complement to Lemma 2. Together, they give a characterization of the linear phase operators with rotation gates that form arbitrary cyclic groups. In particular, the set of phase polynomials which are zero-everywhere mod m for any non-zero $m \in \mathbb{Z}$ is given by scaling the zero-everywhere polynomials for the even-power part of m.

Theorem 5. *Let m be any non-negative integer, and suppose the prime factorization of m is $2^{m_1}3^{m_2}5^{m_3}\cdots$. Then*

$$\mathcal{C}_n^m = \mathcal{C}_n^{2^{m_1}} \cdot 3^{m_2}5^{m_3}\cdots .$$

Proof. Th inclusion of $\mathcal{C}_n^{2^{m_1}} \cdot 3^{m_2}5^{m_3}\cdots$ in $\mathcal{C}_n^m$ is trivial, so let **a** be some tuple in $\mathbb{Z}_m^{2^n-1}$ and suppose $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod d$ for all $\mathbf{x} \in \mathbb{Z}_2^n$.

Clearly $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod d$ for all $\mathbf{x} \in \mathbb{Z}_2^n$ if and only if $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod p_i^{m_i}$ for all $\mathbf{x} \in \mathbb{Z}_2^n$ and prime power $p_i^{m_i}$ in the prime factor decomposition of m. However, by Lemma 3 for any $p \neq 2$, $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod p_i^{m_i}$ if and only if $\mathbf{a} = 0 \mod p_i^{m_i}$, so $\mathbf{a} = \mathbf{a}' \cdot p_i^{m_i}$ where $\mathbf{a}' \in \mathbb{Z}_{m/p_i^{m_i}}$ and $P_{\mathbf{a}'}(\mathbf{x}) = 0 \mod m/p_i^{m_i}$. Repeating for all odd primes, we see that

$$\mathbf{a} = \mathbf{a}' \cdot 3^{m_2}5^{m_3}\cdots$$

for some $\mathbf{a}' \in \mathbb{Z}_{2^{m_1}}$ where $P_{\mathbf{a}'}(\mathbf{x}) = 0 \mod 2^{m_1}$ for all $\mathbf{x} \in \mathbb{Z}_2^n$. Hence $\mathbf{a}' \in \mathcal{C}_n^{2^{m_1}}$ and so $\mathbf{a} \in \mathcal{C}_n^{2^{m_1}} \cdot 3^{m_2}5^{m_3}\cdots$ as required. □

VI. Experiments

We implemented Algorithm 1 in T-par [5] as an optimization pass in the resynthesis procedure. T-par optimizes circuits over the Clifford+T gate set by computing a representation using exponential sums, then resynthesizing. As our algorithm presently applies to CNOT and phase gates, we break up the input circuit into {CNOT, T} subcircuits, each of which is then optimized individually.

We implemented and tested the algorithm with two Reed-Muller decoders – a majority logic decoder due to Reed [27], and a modern recursive decoder due to Dumer [32]. The former has complexity in $O(2^{2n})$ for an n-qubit circuit while the latter has a significantly lower complexity of $O(2^n)$. While both of these algorithms are exponential in the number of qubits n, we nonetheless obtain reasonable performance for large circuits by storing and operating directly on compressed vector representations. In order to optimize these large circuits we chose relatively fast decoders over minimum-distance decoders.

A. Evaluation

Algorithm 1 was evaluated on a suite of benchmark quantum circuits, drawn from the literature and the Reversible Logic Benchmarks page [33]. The majority of circuits tested are reversible circuits, though some specifically quantum circuits were also examined. Toffoli gates were replaced with a Clifford+T implementation using 7 T-gates [3], and multiple control Toffolis were expanded into two-control Toffoli gates using one zero initialized ancilla (see, e.g., [24]).

Table II reports the T-count of circuits optimized with both T-par alone, and with Algorithm 1 using either the majority logic or recursive decoder applied to {CNOT, T} subcircuits. All experiments were run on with a 2.4GHz quad-core Intel Core i7 processor running Linux and 8GB of RAM. Each benchmark had a timeout of 30 minutes – instances where the algorithm failed to report a result within the timeout are identified with a dash.

On average, Algorithm 1 reduced T-count by 6% for both the majority logic decoder and the recursive decoder compared to T-par. While the recursive decoder produced the best results in some cases, notably the Galois field multipliers, and failed less often, for many benchmarks it reported significantly *increased* T-counts compared to T-par. Majority logic decoding by comparison typically produced less T-reduction, though it consistently resulted in circuits with equal or lesser T-count than that reported by T-par. Counter-intuitively this appears to result from the recursive decoder actually doing a *better* job optimizing T-count – after the recursive decoder performs significant rewrites on individual {CNOT, T} subcircuits, T-par has less opportunity to optimize T-gates across subcircuit boundaries. A natural direction of future research is to extend decoding-based optimization to $\{H, \text{CNOT}, T\}$ circuits in order to make use of the additional T-count reductions possible across subcircuit boundaries.

While the T-count reductions over T-par are minor compared to the initial jump from the original T-count, the results clearly demonstrate that further T-count optimization beyond the T-par algorithm is possible. In the most significant case a T-count reduction of 75% was reported for the benchmark BCSD$_8$ with both decoders, though as the benchmark performs state distillation and relies on certain properties of the circuit

[3] Grover's search is performed with 4 iterations using the oracle $f(\mathbf{x}) = \neg x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \wedge \neg x_5$.

TABLE II

T-COUNT OPTIMIZATION RESULTS. n REPORTS THE NUMBER OF QUBITS IN THE CIRCUIT. T-COUNTS ARE RECORDED FOR THE ORIGINAL CIRCUIT, AFTER OPTIMIZATION BY T-PAR, AND AFTER OPTIMIZATION BY ALGORITHM 1 WITH EITHER THE MAJORITY LOGIC OR RECURSIVE DECODER

Benchmark	n	T-count			
		original	T-par	majority	recursive
Grover$_5$ [34][3]	9	140	52	52	52
Mod 5$_4$ [33]	5	28	16	16	16
VBE-Adder$_3$ [35]	10	70	24	24	24
CSLA-MUX$_3$ [36]	15	70	62	62	58
CSUM-MUX$_9$ [36]	30	196	140	84	76
QCLA-Com$_7$ [37]	24	203	95	94	153
QCLA-Mod$_7$ [37]	26	413	249	238	299
QCLA-Adder$_{10}$ [37]	36	238	162	–	188
Adder$_8$ [38]	24	399	215	213	249
RC-Adder$_6$ [39]	14	77	63	47	47
Mod-Red$_{21}$ [40]	11	119	73	73	73
Mod-Mult$_{55}$ [40]	9	49	37	35	35
Mod-Adder$_{1024}$ [33]	28	1995	1011	1011	1011
BCSD$_2$ [41]	9	14	14	2	2
BCSD$_4$ [41]	14	20	20	4	4
BCSD$_8$ [41]	21	32	32	8	8
Cycle 17_3 [33]	35	4739	1945	1944	1982
HWB$_6$ [33]	7	105	75	75	75
HWB$_8$ [33]	12	5887	3551	3531	3531
nth-prime$_6$ [33]	9	812	402	400	400
nth-prime$_8$ [33]	12	6671	4047	4045	4045
GF(2^4)-Mult [42]	12	112	68	68	68
GF(2^5)-Mult [42]	15	175	111	111	101
GF(2^6)-Mult [42]	18	252	150	150	144
GF(2^7)-Mult [42]	21	343	217	217	208
GF(2^8)-Mult [42]	24	448	264	264	237
GF(2^9)-Mult [42]	27	567	351	–	301
GF(2^{10})-Mult [42]	30	700	410	–	410
GF(2^{16})-Mult [42]	48	1792	1040	–	–
GF(2^{32})-Mult [42]	96	7168	4128	–	–
Hamming$_{15}$ (low) [33]	17	161	97	97	97
Hamming$_{15}$ (med) [33]	17	574	230	230	230
Hamming$_{15}$ (high) [33]	20	2457	1019	1019	1019
QFT$_4$ [24]	5	69	67	67	67
$\Lambda_3(X)$ – [43]	5	28	16	16	16
– [24]	5	21	15	15	15
$\Lambda_4(X)$ – [43]	7	56	28	28	28
– [24]	7	35	23	23	23
$\Lambda_5(X)$ – [43]	9	84	40	40	40
– [24]	9	49	31	31	31
$\Lambda_{10}(X)$ – [43]	19	224	100	100	100
– [24]	19	119	71	71	71

for fault tolerance, such an optimization is not likely useful. Note that it may be possible to achieve better T-count with other Reed-Muller decoders as well. We leave exploration of effective decoders as an avenue for future work.

As an additional note, while we do not consider T-depth optimization in this paper, reductions to T-count in some benchmarks allow further reductions to T-depth using matroid partitioning. In the extreme case, T-depth in CSUM-MUX$_9$ was reduced from 11 to 6 using the recursive decoder, providing strong evidence that reducing T-count is an effective means of optimizing T-depth.

VII. CONCLUSION

In this paper we have answered the question previously posed in [4] of whether there exist identities which can be used to reduce the T-cost of a phase polynomial over CNOT and T gates. We gave a concrete set of generators for the entire set of identities and have shown that, when restricted to T-count optimization, these identities correspond exactly to the punctured Reed-Muller code of length $2^n - 1$ and order $n - 4$. From this correspondence we developed a T-count optimization procedure which uses Reed-Muller decoders to reduce the T-cost of a phase polynomial and is optimal when a minimum distance decoder is used, as well as gave a new upper bound on the T-count of $\{\text{CNOT}, T\}$ circuits. We also looked at the question of optimizing phase polynomials corresponding to other Z-basis rotation gates, giving a concrete set of generators for the set of identities over rotations of any finite order.

A natural continuation of this programme is to find methods for minimizing the T-count of quantum circuits over a universal set of gates – for instance, the standard Clifford+T set generated by $\{H, \text{CNOT}, T\}$. Our methods give both an upper bound of $O(k \cdot n^2)$ T-gates for a circuit containing k Hadamard gates, as well as a concrete algorithm which achieves this bound when using a minimum distance decoder. On the other hand, the $(n + k)$-variate phase polynomial for an entire k-Hadamard circuit over $\{H, \text{CNOT}, T\}$ may be computed and optimized directly [4], giving an upper bound of $O((n + k)^2)$ T gates with the caveat that the resulting operator may not be implementable with only n qubits. In either case the minimal T-count depends on the Hadamard cost of the circuit which may itself be reduced, implying that unlike the $\{\text{CNOT}, T\}$ case, the minimal T-count of a Clifford+T circuit may not be achievable simply by rewriting its phase polynomial. We leave it as a question for future research to determine the relationship between phase polynomials, Hadamard gates and ancillas, as well as upper bounds and methods for finding the exact minimal T-count of Clifford+T circuits.

APPENDIX

In this appendix we give an explicit set of generators for the space of zero-everywhere phase polynomials modulo powers of 2. In particular, we give proofs of Lemma 1 and the general version, Lemma 2.

A. The Monomial Basis

Our proof relies on a connection between the binary evaluations of polynomials over $\mathbb{Z}_8$ and the module $\mathbb{Z}_8^{2^n-1}$. In particular, consider the set of degree at most $n - 1$ monomial (Boolean) evaluation vectors

$$\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{1}\}\}.$$

We show that this set of vectors, under the natural inclusion of $\mathbb{Z}_2$ in $\mathbb{Z}_8$, forms a generating set for $\mathbb{Z}_8^{2^n-1}$ – moreover, since each such vector is linearly independent over $\mathbb{Z}_2^{2^n-1}$ and hence also linearly independent over $\mathbb{Z}_8^{2^n-1}$, this set is in fact a basis. We call this basis the *monomial basis* of $\mathbb{Z}_8^{2^n-1}$.

Lemma 4. *$\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{1}\}\}$ is a basis of* $\mathbb{Z}_8^{2^n-1}$

Proof. We first note that the set of all non-constant monomial evaluation vectors, $\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{0}\}\}$, is a basis for the module $\mathbb{Z}_8^{2^n-1}$. In particular, for any

$\mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{0}\}$ the vector $\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ contains a leading 1 at the **y**th index (e.g., Table I), and hence any tuple of $\mathbb{Z}_8^{2^n-1}$ may be written as a linear combination over this set. It therefore suffices to prove that $\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n$ is in the span of $\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{1}\}\}$.

It may be observed that over $\mathbb{Z}_2$, the set of *all* monomial evaluation vectors is linearly dependent, and in particular that

$$\bigoplus_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} = \mathbf{0}$$

since every input evaluates to 1 for an even number of monomials. Further, as Res_2 is homomorphic we have

$$\bigoplus_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} = \mathrm{Res}_2\left(\sum_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right) = \mathbf{0}$$

and so $\sum_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} = \mathbf{a}$ for some $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ such that $\mathrm{Res}_2(\mathbf{a}) = 0$. If we write $\mathbf{a}$ over the basis $\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{0}\}\}$ and move all instances of $\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n$ to the left we see

$$b \cdot \mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n = \mathbf{a}' - \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$$

where $\mathbf{a}'$ is in the span of $\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{0},\mathbf{1}\}\}$ and $b \in \mathbb{Z}_8$.

Now suppose b is even. Then

$$(b-1)\cdot \mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n = \mathbf{a}' - \sum_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$$

Taking the binary residue of both sides we then find

$$\begin{aligned}\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n &= \mathrm{Res}_2(\mathbf{a}') + \mathrm{Res}_2(\sum_{\mathbf{y}\in\mathbb{Z}_2^n} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}) \\ &= \mathrm{Res}_2(\mathbf{a}').\end{aligned}$$

Since $\mathbf{a}'$ is in the span of $\{\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y} \in \mathbb{Z}_2^n\setminus\{\mathbf{0},\mathbf{1}\}\}$ over $\mathbb{Z}_8$, $\mathrm{Res}_2(\mathbf{a}')$ is in its span over $\mathbb{Z}_2$ and hence may be written over this basis. However, the set of all monomial evaluation vectors of degree at least 1 is linearly independent over $\mathbb{Z}_2$, so we arrive at a contradiction.

Thus b is odd and as such has a multiplicative inverse in $\mathbb{Z}_8$. Hence,

$$\begin{aligned}\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n &= b\cdot b^{-1}\cdot \mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n \\ &= b^{-1}\cdot\left(\mathbf{a}' - \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} \mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right).\end{aligned}$$

□

Lemma 4 tells us that any element $\mathbf{a}$ of $\mathbb{Z}_8^{2^n-1}$ is the vector of evaluations for some pseudo-Boolean polynomial function $f : \mathbb{Z}_2^n \to \mathbb{Z}_8$ where

$$f(X_1, X_2, \ldots, X_n) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} b_{\mathbf{y}} X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n},$$

and hence $\mathbf{f} = \mathbf{a} = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ in the monomial basis. Moreover, since

$$\mathrm{Res}_2(\mathbf{a}) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} \mathrm{Res}_2(b_{\mathbf{y}})\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n},$$

$\mathrm{Res}_2(\mathbf{a})$ is the evaluation vector of a Boolean polynomial function with degree at most $\deg(f)$.

B. Evaluating $P_{\mathbf{a}}$

The next step in our proof is to give an analytic formula for the value of a phase function $P_{\mathbf{a}}$ applied to a vector $\mathbf{x} \in \mathbb{Z}_2^n$ as a function of the degree of the polynomial form of $\mathbf{a}$. Specifically, we show that $P_{\mathbf{a}}(\mathbf{x})$ is equal to a linear combination of the Hamming weights – numbers of solutions – of certain Boolean polynomials arising from the multiplication of a monomial with a degree 1 polynomial.

Consider the value of a phase polynomial $P_{\mathbf{a}}$ at $\mathbf{x} \in \mathbb{Z}_2^n$:

$$P_{\mathbf{a}}(\mathbf{x}) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{0}\}} a_{\mathbf{y}}(y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n).$$

We can view the above equation as an inner product, since the value $y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$ is the **y**th component of the evaluation vector $x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_n\mathbf{X}_n$.

Formally, we define $\langle\mathbf{a},\mathbf{b}\rangle$ for $\mathbf{a},\mathbf{b} \in \mathbb{Z}_8^{2^n-1}$ as $\sum_{i=1}^{2^n-1} a_ib_i$. Note that

$$\langle\mathbf{a}+\mathbf{b},\mathbf{c}\rangle = \langle\mathbf{a},\mathbf{c}\rangle + \langle\mathbf{b},\mathbf{c}\rangle$$

for any $\mathbf{a},\mathbf{b},\mathbf{c} \in \mathbb{Z}_8^{2^n-1}$ since the inner product is linear in either argument over $\mathbb{Z}$, and hence also $\mathbb{Z}_8$. Using this observation, we give an explicit formula for $P_{\mathbf{a}}(\mathbf{x})$ as a function of the basis vectors appearing in $\mathbf{a}$:

Lemma 5. *Let $\mathbf{a} \in \mathbb{Z}_8^{2^n}$ and suppose*

$$\mathbf{a} = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$$

in the monomial basis. Then

$$P_{\mathbf{a}}(\mathbf{x}) = \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} b_{\mathbf{y}}|(\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n})(x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots x_n\mathbf{X}_n)|.$$

Proof. By direct calculation.

$$\begin{aligned}
&P_{\mathbf{a}}(\mathbf{x}) \\
&= \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{0}\}} a_{\mathbf{y}}(y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n) \\
&= \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{0}\}} a_{\mathbf{y}}(x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_n\mathbf{X}_n)_{\mathbf{y}} \\
&= \langle\mathbf{a}, x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_n\mathbf{X}_n\rangle \\
&= \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} b_{\mathbf{y}}\langle\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}, x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_n\mathbf{X}_n\rangle \\
&= \sum_{\mathbf{y}\in\mathbb{Z}_2^n\setminus\{\mathbf{1}\}} b_{\mathbf{y}}|(\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n})(x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_n\mathbf{X}_n)|.
\end{aligned}$$

□

The value of

$$|(\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n})(x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_n\mathbf{X}_n)|$$

in Lemma 5 above may be restated as the number of solutions to the equation

$$(X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n})(x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_nX_n) = 1.$$

Fortunately, this number of a simple function of the degree of the polynomial, as the following Lemma shows.

Lemma 6. *Let*

$$f = (X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n})(x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_nX_n)$$

for some $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_2^n$ *– that is,* f *can be factored as the product of a monomial and a linear Boolean polynomial. Then either* $f = 0$*, or* $|\mathbf{f}| = 2^{n-\deg(f)}$.

Proof. Suppose $f \neq 0$. Since $x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_nX_n$ has degree 1, either $\deg(f) = |\mathbf{y}|$ or $\deg(f) = |\mathbf{y}|+1$. We consider these two cases separately.

Consider the degree $|\mathbf{y}|$ case first. Clearly $\mathbf{x} \subseteq \mathbf{y}$ – that is, every variable X_i in the linear combination $X_1x_1 \oplus X_2x_2 \oplus \cdots \oplus X_nx_n$ is already in $X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}$, hence the degree remains unchanged. Then

$$\begin{aligned} f &= |\mathbf{x}|X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n} \\ &= \begin{cases} 0 & \text{if } |\mathbf{x}| = 0 \mod 2 \\ X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n} & \text{otherwise.} \end{cases} \end{aligned}$$

Since $f \neq 0$, we have $|\mathbf{x}| = 1 \mod 2$. Moreover, $X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n} = 1$ has exactly $2^{n-|\mathbf{y}|}$ solutions, corresponding to the valuations where $X_i = 1$ whenever $y_i = 1$.

Now consider the degree $|\mathbf{y}| + 1$ case. We know $\mathbf{x} \not\subseteq \mathbf{y}$. Without loss of generality we can assume $\mathbf{x} \cap \mathbf{y} = \mathbf{0}$ – that is, there are no variables that appear in both $X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}$ and $x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_nX_n$ – as any common variables can be absorbed into the multiplicity of $X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}$ by distributivity.

Recall that $x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_nX_n = 1$ for exactly half of the values of all X_i such that $x_i = 1$. Since $X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n} = 1$ for the $2^{n-|\mathbf{y}|}$ valuations where $X_i = 1$ whenever $y_i = 1$, and $y_i = 1$ implies $x_i = 0$, exactly half of those solutions – $2^{n-|\mathbf{y}|}/2 = 2^{n-(|\mathbf{y}|+1)}$ – satisfy $x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_nX_n = 1$. Hence

$$|\mathbf{f}| = 2^{n-\deg(f)}$$

as required. □

In general, it is not the case that the number of solutions to $f(\mathbf{x}) = 1$ is $2^{n-\deg(f)}$ for an n-variate Boolean polynomial function f. In particular, consider $f = 1 \oplus X_1X_2\cdots X_i$. Since $X_1X_2\cdots X_i = 1$ has 2^{n-i} solutions, the number of solutions to $f(\mathbf{x}) = 1$ is

$$2^n - 2^{n-i} \neq 2^{n-\deg(f)}.$$

C. An Explicit Set of Generators

From Lemma 6 it is immediate that if $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ may be written over the monomial basis with degree at most $n-4$, then $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod 8$ for any $\mathbf{x}$ and so $\mathbf{a} \in \mathcal{C}_n$. However, it may be the case that $\mathbf{a}$ contains monomials with degree greater than $n-4$ and yet are still in $\mathcal{C}_n$. For instance, let $n = 4$ and consider $\mathbf{a} = 2 \cdot \mathbf{X}_1$. Then for any $\mathbf{x} \in \mathbb{Z}_2^4$,

$$\begin{aligned} P_{\mathbf{a}}(\mathbf{x}) &= 2 \cdot |\mathbf{X}_1(x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus \cdots \oplus x_4\mathbf{X}_4)| \\ &= 2 \cdot 2^{4-\deg(X_1(x_1X_1 \oplus x_2X_2 \oplus \cdots \oplus x_4X_4)} \\ &= 0 \mod 8. \end{aligned}$$

In this case we have $\mathbf{a} \in \mathcal{C}_n$ even though as a polynomial over $\mathbb{Z}_8$, $\mathbf{a}$ has degree greater than 0. In particular, with regard to Lemma 6 the term $2 \cdot X_1$ *acts as if it were a term of degree 0*, since for any $\mathbf{x} \in \mathbb{Z}_2^4$,

$$2 \cdot |\mathbf{X}_1(x_1\mathbf{X}_1 \oplus x_2\mathbf{X}_2 \oplus x_3\mathbf{X}_3)| = 0, 2^3 \text{ or } 2^4.$$

With this intuition we define the *order* of a term to be

$$\operatorname{ord}\left(b \cdot X_1^{y_1}X_2^{y_2}\cdots X_n^{y_n}\right) = |\mathbf{y}| - \nu_2(b)$$

where $\nu_2(b)$ is the 2-adic order of b, i.e. the greatest k such that $2^k \mid b$. Moreover, we define the order of a polynomial to be the maximum order of any term. As we show below, the phase polynomial associated with a tuple $\mathbf{a} \in \mathbb{Z}_8^{2^n-1}$ necessarily evaluates to a non-zero value mod 2^k for some input if $\mathbf{a} = \sum_{\mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{1}\}} b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ has order at least $n-k$.

Lemma 7. *Let* $\mathbf{a} \in \mathbb{Z}^{2^n-1}$ *have order* $m > n-k-1$ *in the monomial basis. Then there exists* $\mathbf{x} \in \mathbb{Z}_2^n$ *such that*

$$P_{\mathbf{a}}(\mathbf{x}) \neq 0 \mod 2^k.$$

Proof. Suppose to the contrary that $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod 2^k$ for all $\mathbf{x} \in \mathbb{Z}_2^n$ and let

$$\mathbf{a} = \sum_{\mathbf{y} \in \mathbb{Z}_2^n \setminus \{\mathbf{1}\}} b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}.$$

Since $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod 2^k$ for all $\mathbf{x} \in \mathbb{Z}_2^n$ we must also have $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod 2^{n-m-1}$, as $n-m-1 < n-(n-k-1)-1 = k$. Note that if $\operatorname{ord}\left(b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right) = i$, by Lemmas 5 and 6,

$$P_{b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}}(\mathbf{x}) = 2^{n-i} \text{ or } 2^{n-i-1}.$$

The latter case occurs exactly when $\mathbf{x} \not\subseteq \mathbf{y}$, and hence we see that $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod 2^{n-m-1}$ implies there are *evenly many terms* $b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ *of maximum order* m *such that* $\mathbf{x} \not\subseteq \mathbf{y}$. Alternatively, for any $\mathbf{x} \in \mathbb{Z}_2^n$

$$\begin{aligned} P_{\mathbf{a}}(\mathbf{x}) &= \sum_{\mathbf{y} \in \mathbb{Z}_2^n} P_{b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}}(\mathbf{x}) \\ &= \sum_{\substack{\mathbf{y} \in \mathbb{Z}_2^n, \mathbf{x} \not\subseteq \mathbf{y} \\ \operatorname{ord}\left(b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right) = m}} 2^{n-m-1} \mod 2^{n-m}, \end{aligned}$$

since every other term evaluates either to 2^{n-m} (i.e. if it has order m and $\mathbf{x} \subseteq \mathbf{y}$), or to 2^{n-i} or 2^{n-i-1} where $i < m$. Moreover, we know at least one such term exists, since by Lemma 4 $\mathbf{a}$ has degree at most $n-1$

Our contradiction arises from the fact that there necessarily exists $\mathbf{x} \in \mathbb{Z}_2^n$ such that an odd number of terms

$b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ with order m such that $\mathbf{x}\not\subseteq\mathbf{y}$. In particular, let

$$S_i = \{\mathbf{y}\in\mathbb{Z}_2^n \mid \mathrm{ord}\left(b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right) = m,\, y_i = 0\},$$

that is S_i is the set of terms of maximum order which do not contain X_i. Given some $\mathbf{x}\in\mathbb{Z}_2^n$, $\bigcup_{i|x_i=1} S_i$ gives the set of terms of maximum order such that $\mathbf{x}\not\subseteq\mathbf{y}$, and hence since $P_{\mathbf{a}}(\mathbf{x}) = 0 \mod 2^{n-m-1}$, it follows that

$$|\cup_{i|x_i=1}S_i| = 0 \mod 2.$$

Now take $\mathbf{y}'\in\mathbb{Z}_2^n$ such that $\mathrm{ord}\left(b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right) = m$, minimizing $|\mathbf{y}'|$ – that is, $\mathbf{y}'$ is a term of maximum order but with minimum degree. Since $\mathbf{y}'$ has minimal weight, for every other $\mathbf{y}$ such that $\mathrm{ord}\left(b_{\mathbf{y}}\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}\right) = m$, there necessarily exists i such that $y_i = 1$ but $y_i' = 0$. Hence

$$\cap_{i|y_i'=0}S_i = \{\mathbf{y}'\}.$$

By inclusion-exclusion, the cardinality of this set can be written as a sum of cardinalities of unions of S_i – in particular,

$$\begin{aligned}1 &= |\cap_{i|y_i'=0}S_i| \\ &= \overline{|\cup_{i|y_i'=0}\overline{S_i}|} \\ &= 2^n - |\cup_{i|y_i'=0}\overline{S_i}| \\ &= 2^n - \sum_{k=1}^{|\overline{\mathbf{y}}|}(-1)^{k+1}\left(\sum_{i_1,\ldots,i_k}|\overline{S_{i_1}}\cap\cdots\cap\overline{S_{i_k}}|\right) \\ &= 2^n - \sum_{k=1}^{|\overline{\mathbf{y}}|}(-1)^{k+1}\left(\sum_{i_1,\ldots,i_k}2^n - |S_{i_1}\cup\cdots\cup S_{i_k}|\right)\end{aligned}$$

However, since $|\cup_{i|x_i=1}S_i| = 0 \mod 2$ for any $\mathbf{x}$, we have $1 = 0 \mod 2$, hence we derive our contradiction. □

Lemma 7 suffices to prove that $\mathcal{C}_n^{2^k}$ is generated by the set of scaled monomial vectors of order at most $n-k-1$ – in the case when $k = 3$, corresponding to T-count optimization, we have that $\mathcal{C}_n = \mathcal{C}_n^8$ is generated by terms of order $n-4$. As a consequence we obtain not only a T-count optimization procedure for {CNOT, T} circuits, but also fully characterize the set of diagonal unitaries implementable over {CNOT, T}, in the sense that

$$\mathcal{P}_8(n) \simeq \mathbb{Z}_8^{2^n-1}/\mathcal{C}_n.$$

Lemma 2. *$\mathcal{C}_n^{2^k}$ is generated by*

$$\{2^i\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y}\in\mathbb{Z}_2^n, |\mathbf{y}|-i\le n-k-1\}.$$

Proof. Suppose $\mathbf{c}\in\mathcal{C}_n^{2^k}$. Then $P_{\mathbf{c}}(\mathbf{x}) = 0 \mod 2^k$ for all $\mathbf{x}\in\mathbb{Z}_2^n$, hence by Lemma 7, $\mathbf{c}$ must have order at most $n-k-1$ and can be written as a sum of the above generators.

Now consider some generator $\mathbf{c} = 2^i\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ where $|\mathbf{y}|-i\le n-k-1$. By Lemmas 5 and 6,

$$P_{\mathbf{c}}(\mathbf{x}) = 2^{i+n-|\mathbf{y}|} \text{ or } 2^{i+n-|\mathbf{y}|-1}$$

for any $\mathbf{x}\in\mathbb{Z}_2^n$. Since $i+n-|\mathbf{y}| > i+n-|\mathbf{y}|-1\ge k$ we have $P_{\mathbf{c}}(\mathbf{x}) = 0 \mod 2^k$ so $\mathbf{c}\in\mathcal{C}_n^{2^k}$. Moreover since $\mathcal{C}_n^{2^k}$ is a group, every sum of terms $2^i\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n}$ where $|\mathbf{y}|-i\le n-k-1$ is contained in $\mathcal{C}_n^{2^k}$, hence $\mathcal{C}_n^{2^k}$ is generated by $\{2^i\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y}\in\mathbb{Z}_2^n, |\mathbf{y}|-i\le n-k-1\}$. □

As a corollary to the above we also obtain Lemma 1, namely that $\mathcal{C}_n = \mathcal{C}_n^8$ is generated by

$$\{2^i\mathbf{X}_1^{y_1}\mathbf{X}_2^{y_2}\cdots\mathbf{X}_n^{y_n} \mid \mathbf{y}\in\mathbb{Z}_2^n|\mathbf{y}|-i\le n-4\}.$$

Acknowledgments

The authors would like to thank Adam Paetznick and Neil J. Ross for helpful discussions and comments on an earlier version. IQC and Perimeter Institute are supported in part by the Government of Canada and the Province of Ontario.

References

[1] A. Paetznick and B. W. Reichardt, "Universal fault-tolerant quantum computation with only transversal gates and error correction," *Phys. Rev. Lett.*, vol. 111, nos. 9–30, Aug. 2013, Art. no. 090505. [Online]. Available: https://doi.org/10.1103/PhysRevLett.111.090505

[2] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, no. 1, pp. 2–30, Jan. 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003491602000180

[3] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 32, no. 6, pp. 818–830, Jun. 2013.

[4] M. Amy, D. Maslov, and M. Mosca, "Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 33, no. 10, pp. 1476–1489, Oct. 2014. doi: 10.1109/TCAD.2014.2341953.

[5] M. Amy. *T-Par—A Quantum Circuit Optimizer Based on Sum-Over-Paths Representations*. Accessed: Sep. 17, 2015. [Online]. Available: https://github.com/meamy/t-par

[6] S. Forest, D. Gosset, V. Kliuchnikov, and D. McKinnon, "Exact synthesis of single-qubit unitaries over Clifford-cyclotomic gate sets," *J. Math. Phys.*, vol. 56, no. 8, 2015, Art. no. 082201. doi: 10.1063/1.4927100.

[7] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 34th Annu. Symp. Found. Comput. Sci.*, Nov. 1994, pp. 124–134. doi: 10.1109/SFCS.1994.365700.

[8] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, "An algorithm for the T-count," *Quantum Inf. Comput.*, vol. 14, nos. 15–16, pp. 1261–1276, Nov. 2014. doi: 10.26421/QIC14.15-16.

[9] P. Selinger, "Quantum circuits of T-depth one," *Phys. Rev. A, Gen. Phys.*, vol. 87, no. 4, Apr. 2013, Art. no. 042302. doi: 10.1103/PhysRevA.87.042302.

[10] N. Abdessaied, M. Soeken, and R. Drechsler, "Quantum circuit optimization by Hadamard gate reduction," in *Proc. Int. Conf. Reversible Comput.*, 2014, pp. 149–162. doi: 10.1007/978-3-319-08494-7_12.

[11] D. Maslov, "Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization," *Phys. Rev. A, Gen. Phys.*, vol. 93, no. 2, 2016, Art. no. 022311. doi: 10.1103/PhysRevA.93.022311.

[12] V. Kliuchnikov, D. Maslov, and M. Mosca, "Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates," *Quantum Inf. Comput.*, vol. 13, nos. 7–8, pp. 607–630, 2013. doi: 10.26421/QIC13.7-8.

[13] V. Kliuchnikov, D. Maslov, and M. Mosca, "Asymptotically optimal approximation of single qubit unitaries by Clifford and T circuits using a constant number of ancillary qubits," *Phys. Rev. Lett.*, vol. 110, nos. 9–10, 2013, Art. no. 190502. doi: 10.1103/PhysRevLett.110.190502.

[14] P. Selinger, "Efficient clifford+T approximation of single-qubit operators," *Quantum Inf. Comput.*, vol. 15, nos. 1–2, pp. 159–180, Jan. 2015. doi: 10.26421/QIC15.1-2.

[15] N. J. Ross and P. Selinger, "Optimal ancilla-free Clifford+T approximation of z-rotations," *Quantum Inf. Comput.*, vol. 16, nos. 11–12, pp. 901–953, Sep. 2016. doi: 10.26421/QIC16.11-12.

[16] A. Paetznick and K. M. Svore, "Repeat-until-success: Non-deterministic decomposition of single-qubit unitaries," *Quantum Inf. Comput.*, vol. 14, nos. 15–16, pp. 1277–1301, Nov. 2014. doi: 10.26421/QIC14.15-16.

[17] A. Bocharov, M. Roetteler, and K. M. Svore, "Efficient synthesis of universal repeat-until-success quantum circuits," *Phys. Rev. Lett.*, vol. 114, 2015, Art. no. 080502. doi: 10.1103/PhysRevLett.114.080502.

[18] E. Knill, R. Laflamme, and W. Zurek. (1996). "Threshold accuracy for quantum computation." [Online]. Available: https://arxiv.org/abs/quant-ph/9610011

[19] B. Zeng, A. Cross, and I. L. Chuang, "Transversality versus universality for additive quantum codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6272–6284, Sep. 2011. doi: 10.1109/TIT.2011.2161917.

[20] J. T. Anderson and T. Jochym-O'Connor, "Classification of transversal gates in qubit stabilizer codes," *Quantum Inf. Comput.*, vol. 16, nos. 9–10, pp. 771–802, Jul. 2016. doi: 10.26421/QIC16.9-10.

[21] S. Bravyi and J. Haah, "Magic-state distillation with low overhead," *Phys. Rev. A, Gen. Phys.*, vol. 86, no. 5, 2012, Art. no. 052329. doi: 10.1103/PhysRevA.86.052329.

[22] E. T. Campbell, H. Anwar, and D. E. Browne, "Magic-state distillation in all prime dimensions using quantum Reed–Müller codes," *Phys. Rev. X*, vol. 2, no. 4, Oct./Dec. 2012, Art. no. 041021. doi: 10.1103/PhysRevX.2.041021.

[23] A. J. Landahl and C. Cesare. (2013). "Complex instruction set computing architecture for performing accurate quantum Z rotations with less magic." [Online]. Available: https://arxiv.org/abs/1302.3240

[24] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[25] F. J. MacWilliams and N. J. Sloane, *The Theory of Error Correcting Codes*, vol. 16. Amsterdam, The Netherlands: North Holland, 1978.

[26] D. E. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *IEEE Trans. Comput.* vol. C-3, no. 3, pp. 6–12, Sep. 1954.

[27] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Prof. Group Inf. Theory*, vol. 4, no. 4, pp. 38–49, Sep. 1954.

[28] G. D. Cohen and S. N. Litsyn, "On the covering radius of Reed–Müller codes," *Discrete Math.*, vol. 106, pp. 147–155, Sep. 1992. doi: 10.1016/0012-365X(92)90542-N.

[29] G. Seroussi and A. Lempel, "Maximum likelihood decoding of certain Reed–Müller codes," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 448–450, May 1983. doi: 10.1109/TIT.1983.1056662.

[30] D. Gottesman and I. L. Chuang, "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations," *Nature*, vol. 402, no. 6760, pp. 390–393, Nov. 1999. doi: 10.1038/46503.

[31] G. Duclos-Cianci and D. Poulin, "Reducing the quantum-computing overhead with complex gate distillation," *Phys. Rev. A, Gen. Phys.*, vol. 91, 2015, Art. no. 042315. doi: 10.1103/PhysRevA.91.042315.

[32] I. Dumer, "Recursive decoding and its performance for low-rate Reed–Müller codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 811–823, May 2004.

[33] D. Maslov. (2011). *Reversible logic Synthesis Benchmarks Page*. [Online]. Available: http://webhome.cs.uvic.ca/˜dmaslov/

[34] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, May 1996, pp. 212–219. doi: 10.1145/237814.237866.

[35] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Phys. Rev. A, Gen. Phys.*, vol. 54, p. 147, Jul. 1996. doi: 10.1103/PhysRevA.54.147.

[36] R. Van Meter and K. M. Itoh, "Fast quantum modular exponentiation," *Phys. Rev. A, Gen. Phys.*, vol. 71, no. 5, 2005, Art. no. 052320. doi: 10.1103/PhysRevA.71.052320.

[37] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *Quantum Inf. Comput.*, vol. 6, no. 4, pp. 351–369, Jul. 2006. [Online]. Available: https://doi.org/10.26421/QIC6.4

[38] Y. Takahashi, S. Tani, and N. Kunihiro, "Quantum addition circuits and unbounded fan-out," *Quantum Inf. Comput.*, vol. 10, no. 9, pp. 872–890, Sep. 2010. [Online]. Available: https://doi.org/10.26421/QIC10.9

[39] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. Petrie Moulton. (Oct. 2004). "A new quantum ripple-carry addition circuit." [Online]. Available: https://arxiv.org/abs/quant-ph/0410184

[40] I. L. Markov and M. Saeedi, "Constant-optimized quantum circuits for modular multiplication and exponentiation," *Quantum Inf. Comput.*, vol. 12, nos. 5–6, pp. 361–394, May 2012. doi: 10.26421/QIC12.5-6.

[41] A. G. Fowler, S. J. Devitt, and C. Jones, "Surface code implementation of block code state distillation," *Sci. Rep.*, vol. 3, Jun. 2013, Art. no. 1939. doi: 10.1038/srep01939.

[42] D. Maslov, J. Mathew, D. Cheung, and D. K. Pradhan, "An O(m^2)-depth quantum algorithm for the elliptic curve discrete logarithm problem over GF(2^m)^a," *Quantum Inf. Comput.*, vol. 9, no. 7, pp. 610–621, Jul. 2009. doi: 10.26421/QIC9.7-8.

[43] A. Barenco *et al.*, "Elementary gates for quantum computation," *Phys. Rev. A, Gen. Phys.*, vol. 52, pp. 3457–3467, Nov. 1995.

Matthew Amy is a doctoral candidate in computer science at the University of Waterloo. He previously received the M.Math degree in Quantum Information from the University of Waterloo in 2013, after which he spent two years at the University of Toronto as a research assistant in formal verification, before returning to Waterloo for his doctorate. His research interests span formal methods, programming languages and algebra, with particular interest in their application to quantum computing and quantum circuit design.

Michele Mosca obtained his doctorate in Mathematics in 1999 from the University of Oxford on the topic of Quantum Computer Algorithms. He returned to Waterloo in 1999 as a faculty member. He is a founder of the Institute for Quantum Computing at the University of Waterloo, a Professor in the Department of Combinatorics & Optimization of the Faculty of Mathematics, and a founding member of Waterloos Perimeter Institute for Theoretical Physics. He co-founded evolutionQ Inc. and softwareQ Inc. His current research interests include quantum algorithms and complexity, tools for optimizing the implementation of quantum circuits, and the development of cryptographic tools that will be safe against quantum technologies.

Dr. Mosca's work is published widely in top journals, and he co-authored the respected textbook *An Introduction to Quantum Computing* (OUP). His awards and honors include 2010 Canada's Top 40 Under 40, the Premier's Research Excellence Award (2000-2005), Fellow of the Canadian Institute for Advanced Research (CIFAR) since 2010, Canada Research Chair in Quantum Computation (2002-2012), University Research Chair at the University of Waterloo (2012-present), Queen Elizabeth II Diamond Jubilee Medal (2013), SJU Fr. Norm Choate Lifetime Achievement Award (2017), and a Knighthood (Cavaliere) in the Order of Merit of the Italian Republic (2018).