Fast Quantum Fourier Transforms for a Class of non-abelian Groups

Markus Püschel[1], Martin Rötteler[2], and Thomas Beth[2]

[1] Dept. of Mathematics and Computer Science
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104
pueschel@ece.cmu.edu
[2] Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe
Am Fasanengarten 5
D-76128 Karlsruhe, Germany
{roettele, EISS_Office}@ira.uka.de

Abstract. An algorithm is presented allowing the construction of fast Fourier transforms for any solvable group on a classical computer. The special structure of the recursion formula being the core of this algorithm makes it a good starting point to obtain systematically fast Fourier transforms for solvable groups on a quantum computer. The inherent structure of the Hilbert space imposed by the qubit architecture suggests to consider groups of order 2^n first (where n is the number of qubits). As an example, fast quantum Fourier transforms for all 4 classes of non-abelian 2-groups with cyclic normal subgroup of index 2 are explicitly constructed in terms of quantum circuits. The (quantum) complexity of the Fourier transform for these groups of size 2^n is $O(n^2)$ in all cases.

1 Introduction

Quantum algorithms are a recent subject and possibly of central importance in physics and computer science. It has been shown that there are problems on which a putative quantum computer could outperform every classical computer. A striking example is Shor's factoring algorithm (see [27]).

Here we address a problem used as a subroutine in almost all known quantum algorithms: The quantum Fourier transform (QFT) and its generalization to arbitrary finite groups. In classical computation there exist elaborate methods for the construction of Fourier transforms (e. g., [3], [4], [5], [6], [10], [19]), therefore it is highly interesting to adapt and modify these methods to get a quantum algorithm for the Fourier transform with a much better performance (with respect to the quantum complexity model, see Section 3) than any classical algorithm. First attempts in this direction have been proposed by Beals [2] and Høyer [16]. In this paper we present an algebraic approach using representation theory which can be seen as a first step towards the realization of a large class of generalized Fourier transforms on a quantum computer.

2 Representation Theory and Fourier Transforms

Fourier transforms for finite groups are an interesting and well studied topic for classical computers. We refer to [3], [6], [19], [24] as representatives for a vast number of publications. The reader not familiar with the standard notations concerning group representations should refer to these publications or to standard references like [9] or [26]. For the convenience of the reader we will briefly present the terms and notations from representation theory which we are going to use and recall the definition of Fourier transforms.

A representation of a finite group G of degree $\deg(\phi) = n$ is a homomorphism $\phi : G \to \mathrm{GL}_n(\mathbb{K})$ from G into the group of invertible $(n \times n)$-matrices over a field $\mathbb{K}$. We denote by $1_G : g \mapsto 1$ the *trivial representation* of G (of degree 1). If $A \in \mathrm{GL}_n(\mathbb{K})$, then $\phi^A : g \mapsto A^{-1} \cdot \phi(g) \cdot A$ is the *conjugate* of ϕ by A. ϕ and ψ are called *equivalent*, if $\phi = \psi^A$. If ϕ, ψ are representations of G, then the representation $\phi \oplus \psi : g \mapsto \phi(g) \oplus \psi(g) = \left(\begin{smallmatrix} \phi(g) & 0 \\ 0 & \psi(g) \end{smallmatrix}\right)$ is called the *direct sum* of ϕ and ψ. ϕ is called *irreducible*, if it cannot be conjugated to be a direct sum. In this paper, we will deal only with *ordinary* representations, i. e. the characteristic of $\mathbb{K}$ does not divide the group order $|G|$ (Maschke condition). In this case, every representation ϕ can be conjugated, by a suitable matrix A, to a direct sum of irreducible representations ρ_i (Maschke's theorem), i. e. $\phi^A = \bigoplus_{i=1}^{k} \rho_i$, which is called a *decomposition* of ϕ and A is referred to as a *decomposition matrix* for ϕ. Let ϕ be a representation of $H \leq G$, and $\overline{\phi}$ a representation of G which is equal to ϕ when restricted to H ($\overline{\phi} \downarrow H = \phi$). Then $\overline{\phi}$ is called an *extension* of ϕ to G and ϕ is called *extensible* (to G). Note, that an extension does not exist in general. If ϕ is a representation of $H \leq G$ and $t \in G$ then $\phi^t : h \mapsto \phi(tht^{-1})$ is a representation of H^t, called the *inner conjugate* of ϕ by t. If $H \leq G$ is a subgroup with transversal (i. e. a system of representatives of the right cosets of H in G) $T = (t_1, \ldots, t_k)$, then $(\phi \uparrow_T G)(g) = [\dot{\phi}(t_i g t_j^{-1}) \mid i, j = 1 \ldots n]$, where $\dot{\phi}(x) = \phi(x)$ for $x \in H$ and the all-zero matrix else, is called the *induction* of ϕ to G with transversal T. A regular representation is an induction of the form $\phi = (1_E \uparrow_T G)$ where E denotes the trivial subgroup of G.

Let ϕ be a regular representation of G. A *Fourier transform* for G is any decomposition matrix A of ϕ with the additional property that equivalent irreducibles in the corresponding decomposition are even equal. Note, that the definition says nothing about the transversal fixing ϕ, nor the choice of the irreducible representations. As an example let $G = Z_n = \langle x \mid x^n = 1 \rangle$ be the cyclic group of order n with regular representation $\phi = 1_E \uparrow_T G$, $T = (x^0, x^1, \ldots, x^{n-1})$, and ω_n a primitive nth root of unity. Then $\phi^A = \bigoplus_{i=0}^{n-1} \rho_i$, where $\rho_i : x \mapsto \omega_n^i$ and $A = \mathrm{DFT}_n = \frac{1}{\sqrt{n}}[\omega_n^{ij} \mid i, j = 0 \ldots n-1]$ is the (unitary) discrete Fourier transform well-known from signal processing.

If A is a Fourier transform for the group G, then we will refer to a *fast* Fourier transform as any fast algorithm for the multiplication with A. Of course, the term "fast" depends on the chosen complexity model. Since we are primarily interested in the realization of a fast Fourier transform on a quantum computer (QFT) we first have to define the measure of complexity on this architecture.

3 The Complexity Model used in Quantum Computing

Quantum computing is a topic of recent interest which emerged after the discovery of polynomial algorithms for integer factoring and discrete logarithms by P. Shor (see [27]). The state of a quantum computer is given by a normalised vector in a Hilbert space $\mathcal{H}_n$ of dimension 2^n, which is given the natural tensor structure $\mathcal{H}_n = \mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2$ (n factors). The restriction of the computational space to Hilbert spaces of this particular form is motivated by the idea of a quantum register consisting of n quantum bits. A quantum bit, also called *qubit*, is a state corresponding to one tensor component of $\mathcal{H}_n$ and has the form

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1, \quad \alpha, \beta \in \mathbb{C}.$$

The possible operations this computer can carry out are the elements of the unitary group $\mathcal{U}(2^n)$. To study the complexity of performing unitary operations on n-qubit quantum systems we introduce the following two types of computational primitives (see also [15], this volume): *Local unitary operations* on a qubit i are matrices of the form $U^{(i)} := \mathbf{1}_{2^{i-1}} \otimes U \otimes \mathbf{1}_{2^{n-i}}$, where U is an element of the unitary group $\mathcal{U}(2)$ of 2×2-matrices. Furthermore we need operations which affect two qubits at a time, the standard choice for which is the so-called *controlled NOT gate* (also called measurement gate) between the qubits i (control) and j (target) defined by

$$\mathrm{CNOT}^{(i,j)} := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{pmatrix}$$

when restricted to the tensor component of the Hilbert space spanned by the qubits i and j. We assume that these so-called elementary quantum gates can be performed with cost $O(1)$.

In the graphical notation using quantum wires (for details see [1]) these transforms are written as shown in Figure 1. The lines correspond to the qubits, unaffected qubits are omitted, and the dot $\bullet$ sitting on a wire denotes the control bit.

Fig. 1. Elementary quantum gates

These two types of gates suffice to generate all unitary transformations, which is the content of the following theorem from [1].

Theorem 1. *The set* $\mathcal{G} = \{U^{(i)}, \mathrm{CNOT}^{(i,j)} \mid U \in \mathcal{U}(2),\ i,j = 1 \ldots n,\ i \neq j\}$ *is a generating set for the unitary group* $\mathcal{U}(2^n)$.

This means that for each $U \in \mathcal{U}(2^n)$ there is a word $w_1 w_2 \ldots w_k$ (where $w_i \in \mathcal{G}$ for $i = 1 \ldots k$ is an elementary gate) such that U factorizes as $U = w_1 w_2 \ldots w_k$. In general only exponential upper bounds for the minimal length occuring in factorizations have been proved (see [1]) but there are many interesting classes of unitary matrices in $\mathcal{U}(2^n)$ affording only polylogarithmic word length, which means that the minimal length is asymptotically $O(p(n))$ where p is a polynomial. In the following, we give examples of some particular unitary transforms admitting short factorizations which will be useful in the rest of the paper.

- The symmetric group S_n is embedded in $\mathcal{U}(2^n)$ by the natural operation of S_n on the tensor components (qubits). Let $\tau \in S_n$ and Π_τ the corresponding permutation matrix on 2^n points. Then Π_τ has a $O(n)$ factorization as shown in [22]. As an example in Figure 2 the permutation $(1, 3, 2)$ of the qubits (which corresponds to the permutation $(2, 5, 3)(4, 6, 7)$ on the register) is factored according to $(1, 3, 2) = (1, 2)(2, 3)$.

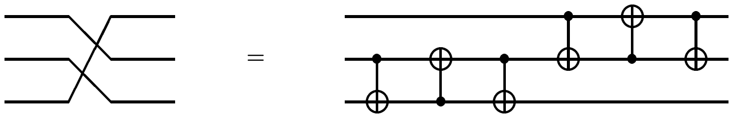

Fig. 2. Factorization $(1, 3, 2) = (1, 2)(2, 3)$

- Following the notation of [1] we denote a k-times controlled U by $\Lambda_k(U)$. As an example for the graphical notation we give in Figure 3 a $\Lambda_1(U)$ gate for arbitrary $U \in \mathcal{U}(2^n)$ with normal, and a gate with inverted control bit including the represented matrix. Lemma 7.2 and Lemma 7.5 in [1] show that for $U \in \mathcal{U}(2)$ the gate $\Lambda_k(U)$ can be realized with gate complexity $O(n)$, for $k < n - 1$, and $\Lambda_{n-1}(U)$ in $O(n^2)$.

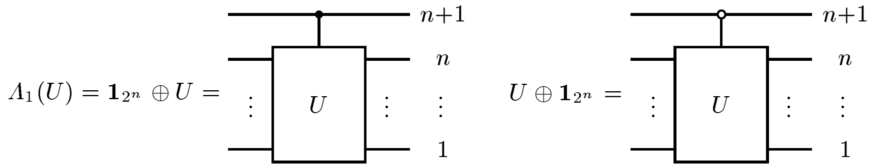

Fig. 3. Controlled gates with a) normal and b) inverted control bit

- The Fourier transform DFT_{2^n} can be performed in $O(n^2)$ elementary operations on a quantum computer (see [27], [8]).
- Let $P_n \in S_{2^n}$ be the cyclic shift acting on the states of the quantum register as $x \mapsto x + 1 \bmod 2^n$. The corresponding permutation matrix is the 2^n-cycle $(1, 2, \ldots, 2^n)$. P_n can be realized using $O(n^2)$ basic operations (see [13], Section 4.4).

Let $U \in \mathcal{U}(2^n)$. The cost for a (single) controlled U is settled by the following

Lemma 1. *If $U \in \mathcal{U}(2^n)$ can be realized in $O(p(n))$ elementary operations then $\Lambda_1(U) \in \mathcal{U}(2^{n+1})$ can also be realized in $O(p(n))$ basic operations.*

Proof: First we assume without loss of generality that U is written in elementary gates. Therefore we have to show that a double controlled NOT and a single controlled $U \in \mathcal{U}(2)$ can be realized with a constant increase of length. This follows from [1]. □

4 Creating Fast Fourier Transforms

In Section 2 we have explained that a Fourier transform for a group G is a decomposition matrix B for a regular representation ϕ of G with the additional property that equivalent irreducible summands are equal, i. e.

$$\phi^B = \rho_1 \oplus \ldots \oplus \rho_k \quad \text{fulfilling} \quad \rho_i \cong \rho_j \Rightarrow \rho_i = \rho_j.$$

A "fast" Fourier transform (on a classical computer) is given by a factorization of B into a product of sparse matrices. (see [3], [6], [19], [25]). For a solvable group G, this factorization can be obtained recursively using the following idea. First, a normal subgroup of prime index $(G : N) = p$ is chosen. Using transitivity of induction, $\phi = 1_E \uparrow G$ is written as $(1_E \uparrow N) \uparrow G$ (note, that we have the freedom to choose the transversals appropriately). Then $1_E \uparrow N$, which again is a regular representation, is decomposed (by recursion) yielding a Fourier transform A for N. In the last step, B is derived from A using a recursion formula.

In the following, we will explain this procedure in more detail by first presenting the two essential theorems (without proof) and then stating the actual algorithm for deriving fast Fourier transforms for solvable groups. The special tensor structure of the recursion formula mentioned above will allow us to use the very same algorithm as a starting point to also obtain fast *quantum* Fourier transforms in the case G being a 2-group (i. e. $|G|$ is a power of 2).

The statements in this section are all taken from the first chapter of [24] where decomposition matrices for arbitrary monomial representations in general are investigated. The first thing we need is Clifford's Theorem which explains the relationship between the irreducible representations of N and those of G.

Theorem 2 (Clifford). *Let $N \trianglelefteq G$ be a normal subgroup of prime index p with (cyclic) transversal $T = (t^0, t^1, \ldots, t^{(p-1)})$ and denote by $\lambda_i : \;\; t \mapsto \omega_p^i$, $i = 0 \ldots p-1$, the p irreducible representations of G arising from G/N. Assume ρ is an irreducible representation of N. Then exactly one of the two following cases applies:*

1. *$\rho \cong \rho^t$ and ρ has p pairwise inequivalent extensions to G. If $\overline{\rho}$ is one of them, then all are given by $\lambda_i \cdot \overline{\rho}$, $i = 0 \ldots p-1$.*
2. *$\rho \ncong \rho^t$ and $\rho \uparrow_T G$ is irreducible. Furthermore, $(\rho \uparrow_T G) \downarrow N = \bigoplus_{i=0}^{p-1} \rho^{t^i}$ and*

$$(\lambda_i \cdot (\rho \uparrow_T G))^{D \otimes \mathbf{1}_d} = \rho \uparrow_T G, \quad D = \operatorname{diag}(1, \omega_p, \ldots, \omega_p^{(p-1)})^i.$$

The following theorem provides the recursion formula which had already been used in [3] to obtain fast Fourier transforms.

Theorem 3. *Let $N \trianglelefteq G$ be a normal subgroup of prime index p with transversal $T = (t^0, t^1, \ldots, t^{(p-1)})$ and ϕ a representation of degree d of N. Suppose that A is matrix decomposing ϕ into irreducibles, i. e. $\phi^A = \rho = \rho_1 \oplus \ldots \oplus \rho_k$ and that $\overline{\rho}$ is an extension of ρ to G. Then*

$$(\phi \uparrow_T G)^B = \bigoplus_{i=0}^{p-1} \lambda_i \cdot \overline{\rho},$$

where $\lambda_i : \; t \mapsto \omega_p^i$, $i = 0 \dots p-1$, *are the* p *irreducible representations of* G *arising from the factor group* G/N,

$$B = (\mathbf{1}_p \otimes A) \cdot D \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_d), \quad \textit{and} \quad D = \bigoplus_{i=0}^{p-1} \overline{\rho}(t)^i.$$

If, in particular, $\overline{\rho}$ *is a direct sum of irreducibles, then* B *is a decomposition matrix of* $\phi \uparrow_T G$.

In case of an cyclic group G the formula yields exactly the well-known Cooley-Tukey decomposition, [7], in which D is usually called the *Twiddle matrix*.

Assume that $N \trianglelefteq G$ is a normal subgroup of prime index p with Fourier transform A and decomposition $\phi^A = \rho = \bigoplus_{i=1}^m \rho_i$. We can reorder the ρ_i, such that the first, say k, ρ_i have an extension $\overline{\rho}_i$ to G and the other ρ_i occur as sequences $\rho_i \oplus \rho_i^t \oplus \dots \oplus \rho_i^{t^{(p-1)}}$ of inner conjugates (cf. Theorem 2, note that irreducibles ρ_i, $\rho_i^{t^j}$ have the same multiplicity since ϕ is regular). In the first case the extension may be calculated by Minkwitz' formula, [21], in the latter case each sequence can be extended by $\rho_i \uparrow_T G$ (Theorem 2, Case 2). We do not state Minkwitz' formula here, since we will not need it in the special cases treated in Section 5. Altogether we obtain an extension $\overline{\rho}$ of ρ and can apply Theorem 3. The remaining task is to assure, that equivalent irreducibles in $\bigoplus_{i=1}^p \lambda_i \cdot \overline{\rho}$ are equal. For summands of $\overline{\rho}$ of the form $\overline{\rho}_i$ we have that $\lambda_j \cdot \overline{\rho}_i$ and $\overline{\rho}_i$ are inequivalent and hence there is nothing to do. For summands of $\overline{\rho}$ of the form $\rho_i \uparrow_T G$, we conjugate $\lambda_j \cdot (\rho_i \uparrow_T G)$ onto $\rho_i \uparrow_T G$ using Theorem 2, Case 2.

Now we are ready to formulate the recursive algorithm for constructing a fast Fourier transform for a solvable group G.

Algorithm 1. Let $N \trianglelefteq G$ a normal subgroup of prime index p with transversal $T = (t^0, t^1, \dots, t^{(p-1)})$. Suppose that ϕ is a regular representation of N with (fast) Fourier transform A, i. e. $\phi^A = \rho_1 \oplus \dots \oplus \rho_k$, fulfilling $\rho_i \cong \rho_j \Rightarrow \rho_i = \rho_j$. A Fourier transform B of G with respect to the regular representation $\phi \uparrow_T G$ can be obtained as follows.

1. Determine a permutation matrix P rearranging the ρ_i, $i = 1 \dots k$, such that the extensible ρ_i (i. e. those satisfying $\rho_i = \rho_i^t$) come first followed by the others ordered into sequences of length p equivalent to $\rho_i, \rho_i^t, \dots, \rho_i^{t^{(p-1)}}$. (Note: These sequences need to be equal to $\rho_i, \rho_i^t, \dots, \rho_i^{t^{(p-1)}}$ which is established in the next step).
2. Calculate a matrix M which is the identity on the extensibles and conjugates the sequences of length p to make them equal to $\rho_i, \rho_i^t, \dots, \rho_i^{t^{(p-1)}}$.
3. Note that $A \cdot P \cdot M$ is a decomposition matrix for ϕ, too, and let $\rho = \phi^{A \cdot P \cdot M}$. Extend ρ to G summandwise. For the extensible summands use Minkwitz' formula, the sequences $\rho_i, \rho_i^t, \dots, \rho_i^{t^{(p-1)}}$ can be extended by $\rho_i \uparrow_T G$.
4. Evaluate $\overline{\rho}$ at t and build $D = \bigoplus_{i=0}^{p-1} \overline{\rho}(t)^i$.

5. Construct a blockdiagonal matrix C with Theorem 2, Case 2, conjugating $\bigoplus_{i=0}^{p-1} \lambda_i \cdot \overline{\rho}$ such that equivalent irreducibles are equal. C is the identity on the extended summands.

Result:

$$B = (\mathbf{1}_p \otimes A \cdot P \cdot M) \cdot D \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_{|N|}) \cdot C \qquad (1)$$

is a fast Fourier transform for G. □

It is obviously possible to construct fast Fourier transforms on a classical computer for any solvable group by recursive use of this algorithm.

Since we restrict ourselves to the case of a quantum computer consisting of qubits, i. e. two-level systems, we apply Algorithm 1 to obtain QFTs for 2-groups ($|G| = 2^n$, $p = 2$). In this case the two tensor products occuring in (1) fit very well to yield a coarse factorization as shown in Figure 4. The lines correspond to the qubits like in Section 3 and a box ranging over more than one line denotes a matrix admitting no a priori factorization into a tensor product.

The remaining problem is the realization of the matrices A, P, M, D, C in terms of elementary building blocks as presented in Section 3. At present, however, this realization remains a creative process which might be performed by hand if a certain class of groups is given. In Section 5 we will apply Algorithm 1 to a class of non-abelian 2-groups.

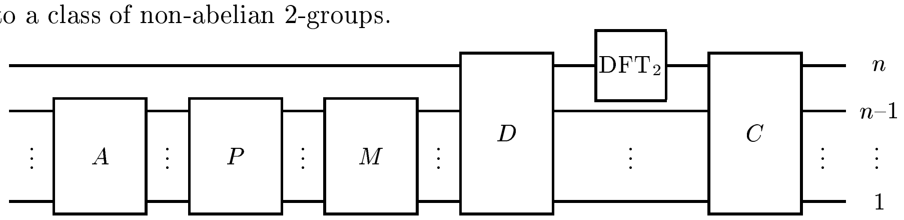

Fig. 4. Coarse quantum circuit visualizing Algorithm 1 for 2-groups

5 Generating QFTs for a class of 2-groups

In the case of G being an abelian 2-group the realization of a fast quantum Fourier transform has been settled by [18]. This case is also covered by the method presented here. In this section we will apply Algorithm 1 to the class of non-abelian 2-groups containing a cyclic normal subgroup of index 2. Fast quantum Fourier transforms for these groups have already been constructed by Høyer in [16].

According to [17], p. 90/91, there are for $n \geq 3$ exactly four isomorphism types of non-abelian groups of order 2^{n+1} affording a cyclic normal subgroup of order 2^n:

1. The dihedral group $D_{2^{n+1}} = \langle x, y \mid x^{2^n} = y^2 = 1,\ x^y = x^{-1} \rangle$.
2. The quaternion group $Q_{2^{n+1}} = \langle x, y \mid x^{2^n} = y^4 = 1,\ x^y = x^{-1} \rangle$.
3. The group $QP_{2^{n+1}} = \langle x, y \mid x^{2^n} = y^2 = 1,\ x^y = x^{2^{n-1}+1} \rangle$.
4. The quasi-dihedral group $QD_{2^{n+1}} = \langle x, y \mid x^{2^n} = y^2 = 1,\ x^y = x^{2^{n-1}-1} \rangle$.

Observe that the extensions 1, 3, and 4 of the cyclic subgroup $Z_{2^n} = \langle x \rangle$ split, i. e. the groups have the structure of a semidirect product of Z_{2^n} by Z_2. The three isomorphism types correspond to the three different embeddings of $Z_2 = \langle y \rangle$ into $(Z_{2^n})^\times \cong Z_2 \times Z_{2^{n-2}}$.

5.1 QFT for the dihedral groups $\mathbf{D_{2^{n+1}}}$

In this section we construct a QFT for the dihedral groups $D_{2^{n+1}}$ step by step according to Algorithm 1 and explicitly state the occuring quantum circuits.

Let $G = D_{2^{n+1}} = \langle x, y \mid x^{2^n} = y^2 = 1,\ x^y = x^{-1} \rangle$ with normal subgroup $N = \langle x \rangle \trianglelefteq G$ of index 2 and transversal $T = (1, y)$. We consider the regular representation $\phi = (1_E \uparrow_S N) \uparrow_T G$ of G with $S = (1, x, \ldots, x^{2^n - 1})$. Obviously the regular representation $(1_E \uparrow_S N)$ of N is decomposed by $A = \mathrm{DFT}_{2^n}$ into $\rho_0 \oplus \ldots \oplus \rho_{2^n - 1}$ where $\rho_i :\ x \mapsto \omega_{2^n}^i$. Now we are ready to apply Algorithm 1 to obtain a decomposition matrix B for ϕ. For convenience we denote ω_{2^n} simply as ω and $H = \mathrm{DFT}_2 = \frac{1}{\sqrt{2}} \cdot \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$.

1. Since $\rho_i^y(x) = \rho_i(yxy^{-1}) = \rho_i(x^{-1}) = \rho_{2^n - i}(x)$ we see that there are exactly two extensible ρ_i namely for $i = 0, 2^{n-1}$. The sequences of inner conjugates are given by $\rho_i, \rho_{2^n - i}$, $i \neq 0, 2^{n-1}$. Thus we need a permutation P reordering the ρ_i as
$$\underbrace{\rho_0, \rho_{2^{n-1}}}_{\text{extensibles}},\ \underbrace{\rho_1, \rho_{2^n - 1}, \ldots, \rho_i, \rho_{2^n - i}, \ldots, \rho_{2^{n-1}-1}, \rho_{2^{n-1}+1}}_{\text{pairs of inner conjugates}}.$$
This can be accomplished by the circuit given in Figure 5 since the n-cycle on the qubits which is performed first yields a decimation by two on the indices, i. e. the indices $0, \ldots, 2^{n-1} - 1$ have found their correct position. The only thing which remains to do is to perform the operation $x \mapsto -x$ on the odd positions. This can be done by an inversion of all (odd) bits followed by a $x \mapsto x + 1$ shift P_{n-1} on the odd states of the register.

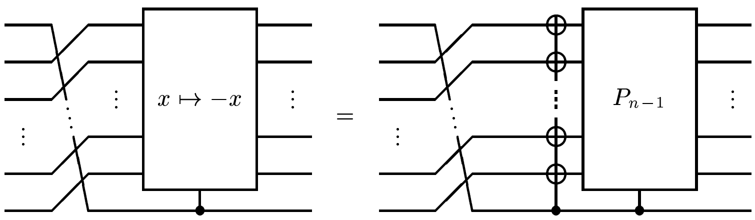

Fig. 5. Ordering the irreducibles of $Z_{2^n} \trianglelefteq D_{2^{n+1}}$

2. M can be omitted since all the ρ_i are of degree 1.
3. Let $\phi^{A \cdot P} = \rho$. We extend ρ summandwise to $\overline{\rho}$:
   - $\rho_0 = 1_N$ can be extended by 1_G.
   - $\rho_{2^{n-1}}$ can be extended through $\overline{\rho}_{2^{n-1}}(y) = 1$.
   - The sequences $\rho_i \oplus \rho_{2^n - i}$, $i \neq 0, 2^{n-1}$ can be extended by $\rho_i \uparrow_T G$ and $(\rho_i \uparrow_T G)(y) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$.

4. Evaluation of $\overline{\rho}$ at the transversal T yields the Twiddle matrix

$$D = \overline{\rho}(1) \oplus \overline{\rho}(y) = \mathbf{1}_{2^n} \oplus \mathbf{1}_2 \oplus \underbrace{\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right) \oplus \ldots \oplus \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)}_{2^{n-1}-1},$$

   which is realized by the quantum circuit given in Figure 6.

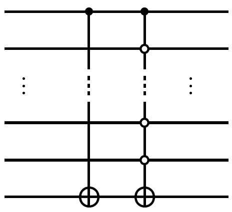

Fig. 6. Twiddle matrix for $D_{2^{n+1}}$

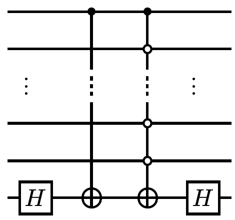

Fig. 7. Equalizing inductions

5. According to Theorem 2, Case 2, the matrix C has the following diagonal form:

$$C = \mathbf{1}_{2^n} \oplus \operatorname{diag}(1, 1, \underbrace{1, -1, \ldots, 1, -1}_{2^{n-1}-1 \text{ pairs}}),$$

   which is realized by the quantum circuit given in Figure 7.

We obtain that $B = (\mathbf{1}_p \otimes A \cdot P \cdot M) \cdot D \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_{|N|}) \cdot C$ is a decomposition matrix for ϕ and a fast quantum Fourier transform for G. The whole circuit is shown in Figure 8.

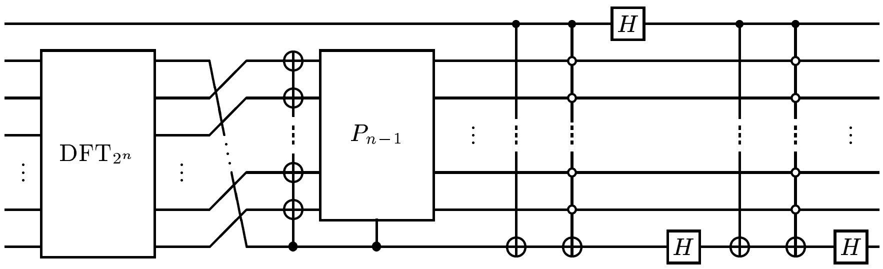

Fig. 8. Complete QFT circuit for the dihedral group $D_{2^{n+1}}$

5.2 QFT for the groups $\mathbf{Q_{2^{n+1}}}$, $\mathbf{QP_{2^{n+1}}}$, and $\mathbf{QD_{2^{n+1}}}$

In the following we give the circuits for the groups $Q_{2^{n+1}}$, $QP_{2^{n+1}}$, and $QD_{2^{n+1}}$. In all cases we have $\langle x \rangle = N \trianglelefteq G$ so that Algorithm 1 has to be performed only once for the last step. For the sake of brevity we will state only those parts of the circuit which differ from the dihedral group. We use the same notation as in the last section.

$Q_{2^{n+1}}$: The irreducibles ρ_i extend or induce in the same way as in the dihedral case. Hence the QFT only differs in the Twiddle matrix D since for a not extensible ρ_i we have $(\rho_i \uparrow_T G)(y) = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. Thus D is given by

$$D = \mathbf{1}_{2^n} \oplus \mathbf{1}_2 \oplus \underbrace{\left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right) \oplus \ldots \oplus \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)}_{2^{n-1}-1}$$

and can be realized by the circuit in Figure 9.

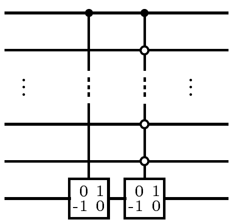

Fig. 9. Twiddle matrix for $Q_{2^{n+1}}$

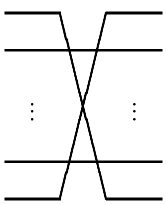

Fig. 10. Permutation for $QP_{2^{n+1}}$

$QP_{2^{n+1}}$: To determine which ρ_i are extensible we observe $\rho_i^y(x) = \rho_i(yxy^{-1}) = \rho_i(x^{2^{n-1}+1})$. Hence $\rho_i = \rho_i^y \Leftrightarrow \omega^i = \omega^{i\cdot(2^{n-1}+1)} \Leftrightarrow \omega^{i\cdot 2^{n-1}} = 1 \Leftrightarrow 2 \mid i$ and there are exactly 2^{n-1} extensible ρ_i. The reordering permutation P has the easy form shown in Figure 10, and the matrix D is given by

$$D = \mathbf{1}_{2^n} \oplus \mathbf{1}_{2^{n-1}} \oplus \underbrace{\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right) \oplus \ldots \oplus \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)}_{2^{n-2}},$$

which is simply a double controlled NOT as visualized in Figure 11. The matrix C then is given by Figure 12.

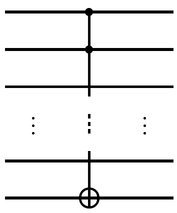

Fig. 11. Twiddle matrix for $QP_{2^{n+1}}$

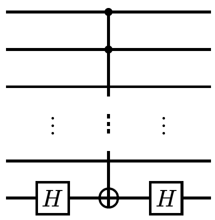

Fig. 12. Equalizing for $QP_{2^{n+1}}$

$QD_{2^{n+1}}$: Here we have $\rho_i^y(x) = \rho_i(x^{2^{n-1}-1})$ and

$$\rho_i = \rho_i^y \Leftrightarrow \omega^i = \omega^{i\cdot(2^{n-1}-1)} \Leftrightarrow \omega^{i\cdot(2^{n-1}-2)} = 1 \Leftrightarrow i = 0, 2^{n-1}.$$

Thus everything is the same as in the dihedral case beside the ordering permutation P which takes the more complicated form shown in Figure 13.

Concerning the complexity of these QFTs we have the following theorem.

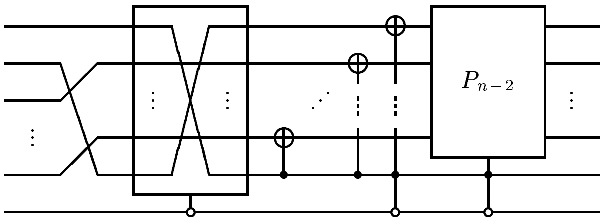

Fig. 13. The permutation for the $QD_{2^{n+1}}$

Theorem 4. *The Fourier transforms for the groups* $G = D_{2^n}$, Q_{2^n}, QP_{2^n}, *and* QD_{2^n} *can be performed on a quantum computer in* $O(\log^2 |G|)$ *elementary operations.*

Proof: We can treat the four series uniformly, since the Fourier transforms all have the same decomposition pattern. First, in all cases a Fourier transform for the normal subgroup $Z_{2^{n-1}}$ is performed with cost of $O(n^2)$ basic operations. The reordering permutation P, the Twiddle matrix D, and the equalizing matrix C cost $O(n^2)$ in case of D_{2^n}, Q_{2^n}, and QD_{2^n} due to Lemma 1 and the examples in Section 3. For QP_{2^n} we need only $O(1)$ operations for P, D, and C. □

All presented Fourier transforms have been implemented by the authors in the language GAP [14] using the share package AREP [12].

6 Conclusions and Outlook

A constructive algorithm has been presented allowing to attack the problem of constructing fast Fourier transforms for 2-groups G on a quantum computer built up from qubits. For a certain class of non-abelian 2-groups the algorithm has been successfully applied. All the QFTs created are of computational complexity $O(\log^2 |G|)$ like in the case of the cyclic group Z_{2^n}. The main problem imposed by the implementation of certain permutation and block diagonal matrices has been solved efficiently.

Using the recursion formula from Theorem 3 it should be possible to construct QFTs for other classes of groups as well as to realize certain signal transforms on a quantum computer by means of symmetry-based decomposition (see [24], [11], [20]).

Acknowledgments

The authors are indebted to Markus Grassl for helpful comments and fruitful discussions. Part of this work was presented and completed during the 1998 Elsag-Bailey – I.S.I. Foundation research meeting on quantum computation. M. R. is supported by Deutsche Forschungsgemeinschaft, Graduiertenkolleg Beherrschbarkeit Komplexer Systeme under Contract No. DFG-GRK 209/3-98.

References

1. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, November 1995.

2. R. Beals. Quantum computation of Fourier transforms over the symmetric groups. In *Proc. STOC 97*, El Paso, Texas, 1997.
3. Th. Beth. *Verfahren der Schnellen Fouriertransformation*. Teubner, 1984.
4. Th. Beth. On the computational complexity of the general discrete Fourier transform. *Theoretical Computer Science*, 51:331–339, 1987.
5. M. Clausen. Fast generalized Fourier transforms. *Theoretical Computer Science*, 67:55–63, 1989.
6. M. Clausen and U. Baum. *Fast Fourier Transforms*. BI-Verlag, 1993.
7. James W. Cooley and John W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19:297–301, 1965.
8. D. Coppersmith. An Approximate Fourier Transform Useful for Quantum Factoring. Technical Report RC 19642, IBM Research Division, 1994.
9. W.C. Curtis and I. Reiner. *Methods of Representation Theory*, volume 1. Interscience, 1981.
10. P. Diaconis and D. Rockmore. Efficient computation of the Fourier transform on finite groups . *Amer. Math. Soc.*, 3(2):297–332, 1990.
11. S. Egner. *Zur Algorithmischen Zerlegungstheorie Linearer Transformationen mit Symmetrie*. PhD thesis, Universität Karlsruhe, Informatik, 1997.
12. S. Egner and M. Püschel. AREP – *A Package for Constructive Representation Theory and Fast Signal Transforms*. GAP share package, 1998. `http://avalon.ira.uka.de/home/pueschel/arep/arep.html`.
13. A. Fijany and C. P. Williams. Quantum Wavelet Transfoms: Fast Algorithms and Complete Circuits. In *Proc. NASA conference QCQC 98*, LNCS vol. 1509, 1998.
14. The GAP Team, Lehrstuhl D für Mathematik, RWTH Aachen, Germany and School of Mathematical and Computational Sciences, U. St. Andrews, Scotland. GAP – *Groups, Algorithms, and Programming, Version 4*, 1997.
15. M. Grassl, W. Geiselmann, and Th. Beth. Quantum Reed–Solomon Codes. In *Proc. of the AAECC–13 (this volume)*, 1999.
16. P. Høyer. Efficient Quantum Transforms. LANL preprint quant–ph/9702028, February 1997.
17. B. Huppert. *Endliche Gruppen*, volume I. Springer, 1983.
18. A. Yu. Kitaev. Quantum Measurements and the Abelian Stabilizer Problem. LANL preprint quant–ph/9511026, November 1995.
19. D. Maslen and D. Rockmore. Generalized FFTs – A survey of some recent results. In *Proceedings of IMACS Workshop in Groups and Computation*, volume 28, pages 182–238, 1995.
20. T. Minkwitz. *Algorithmensynthese für lineare Systeme mit Symmetrie*. PhD thesis, Universität Karlsruhe, Informatik, 1993.
21. T. Minkwitz. Extension of Irreducible Representations. *AAECC*, 7:391–399, 1996.
22. C. Moore and M. Nilsson. Some notes on parallel quantum computation. LANL preprint quant–ph/9804034, April 1998.
23. M. Püschel. *Konstruktive Darstellungstheorie und Algorithmengenerierung*. PhD thesis, Universität Karlsruhe, Informatik, 1998. Translated in [24].
24. M. Püschel. Constructive representation theory and fast signal transforms. Technical Report Drexel-MCS-1999-1, Drexel University, Philadelphia, 1999. Translation of [23].
25. D. Rockmore. Some applications of generalized FFT's. In *Proceedings of DIMACS Workshop in Groups and Computation*, volume 28, pages 329–370, 1995.
26. J. P. Serre. *Linear Representations of Finite Groups*. Springer, 1977.
27. P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithm and Factoring. In *Proc. FOCS 94*, pages 124–134. IEEE Computer Society Press, 1994.