On the Irresistible Efficiency of Signal Processing Methods in Quantum Computing

Andreas Klappenecker*1,2, Martin Rötteler†2
1Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA
2Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe,‡
   Am Fasanengarten 5, D-76128 Karlsruhe, Germany

March 31, 2022

Abstract

We show that many well-known signal transforms allow highly efficient realizations on a quantum computer. We explain some elementary quantum circuits and review the construction of the Quantum Fourier Transform. We derive quantum circuits for the Discrete Cosine and Sine Transforms, and for the Discrete Hartley transform. We show that at most $O(\log^2 N)$ elementary quantum gates are necessary to implement any of those transforms for input sequences of length $N$.

§1 Introduction

Quantum computers have the potential to solve certain problems at much higher speed than any classical computer. Some evidence for this statement is given by Shor’s algorithm to factor integers in polynomial time on a quantum computer. A crucial part of Shor’s algorithm depends on the discrete Fourier transform. The time complexity of the quantum Fourier transform is polylogarithmic in the length of the input signal. It is natural to ask whether other signal transforms allow for similar speed-ups.

We briefly recall some properties of quantum circuits and construct the quantum Fourier transform. The main part of this paper is concerned with

*e-mail: klappi@ira.uka.de
†e-mail: roettele@ira.uka.de
‡research group Quantum Computing, Professor Thomas Beth
the construction of quantum circuits for the discrete Cosine transforms, for the discrete Sine transforms, and for the discrete Hartley transform.

§2 Elementary Quantum Circuits

The quantum computation will be done in the state space of \( n \) two-level quantum systems, which is given by a \( 2^n \)-dimensional complex vector space. The basis vectors are denoted by \(|x\rangle\) where \( x \) is a binary string of length \( n \). The basic unit of quantum information processing is a quantum bit or shortly qubit, which represents the state of a two-level quantum system.

A quantum gate on \( n \) qubits is an element in the group of unitary matrices \( U(2^n) \). There are two types of gates that are considered elementary: the XOR gates (also known as controlled NOTs) and the single qubit operations.

The controlled NOT gate operates on two qubits. It negates the target qubit if and only if the control qubit is 1. Suppose that \( x = b_n \ldots b_1 \) is a string of \( n \) bits, then

\[
\text{CNOT}_{c,t} |x\rangle = \begin{cases} |y\rangle & \text{if } b_c = 1 \\ |x\rangle & \text{if } b_c = 0 \end{cases}
\]

where \( y \) is the bitstring obtained from \( x \) by negating the bit \( b_t \).

A single qubit gate acts on a target qubit at position \( t \) by a local unitary transformation

\[
1_{2^{n-t}} \otimes U \otimes 1_{2^t-1}, \quad U \in U(2).
\]

It will be convenient to describe the quantum circuits with a graphical notation put forward by Feynman. The circuits are read from left to right like a musical score. The qubits are represented by lines, with the most significant bit at the top. Figure 1 shows the graphical notation of the elementary gates.

**Figure 1:** The Feynman notation for the single qubit gate \((1_2 \otimes U)\) and for a controlled NOT operation \(|b_2 b_1\rangle = |b_2 b_2 \oplus b_1\rangle\).

A multiply controlled NOT is defined as follows. Let \( C \) be a subset of \([1..n]\) not containing the target \( t \). Then

\[
\text{CNOT}_{C,t} |x\rangle = \begin{cases} |y\rangle & \text{if } b_c = 1 \text{ for all } c \in C \\ |x\rangle & \text{otherwise} \end{cases}
\]
where $|y\rangle$ is defined as above. Several controlled NOT operations in a sequence allow us to implement the operation $P_n |x\rangle = |x + 1 \mod 2^n\rangle$. Note that $O(n)$ elementary gates are sufficient to realize a multiply controlled NOT operation on $n$ qubits, assuming that an additional scratch qubit is available. Therefore, at most $O(n^2)$ elementary gates are necessary to implement the shift operation $P_n$.

![Diagram of shift operation](image)

**Figure 2: Shift**

The state of two qubits can be exchanged with the help of three controlled NOT operations:

$$\text{SWAP}_{k,h} = \text{CNOT}_{h,k} \text{CNOT}_{k,h} \text{CNOT}_{h,k}.$$  

It follows that any permutation of the $n$ quantum wires can be realized with at most $O(n)$ elementary quantum gates.

A more detailed discussion of properties of quantum gates can be found in [1]. We will discuss the construction of the discrete Fourier transform in the next section. In particular, we will show the classical dataflow diagram and the corresponding quantum gates to further illustrate the graphical notation.

## §3 Quantum Fourier Transform

The discrete Fourier transform of length $N = 2^n$ is defined by

$$F_N = \frac{1}{\sqrt{N}} \left[ \omega^{jk} \right]_{j,k=0,\ldots,N-1},$$

where $\omega = \exp(2\pi i/N)$ with $i^2 = -1$. Recall the recursion step used in the Cooley-Tukey decomposition:

$$F_N = \Upsilon_N (I_2 \otimes F_{N/2}) \begin{pmatrix} 1_{N/2} & T_{N/2} \end{pmatrix} (F_2 \otimes 1_{N/2}) \quad (1)$$
where \( T_{N/2} := \text{diag}(1, \omega, \omega^2, \ldots, \omega^{N/2-1}) \) denotes the matrix of twiddle factors, and \( \Upsilon_N \) denotes the permutation given by \( \Upsilon_N |xb\rangle = |bx\rangle \) with \( x \) an \( n - 1 \)-bit integer, and \( b \) a single bit.

We note that the implementation of \( F_2 \) is a local operation on a single quantum bit. The recursion suggest four different parts of the implementation of Fourier transforms of larger length. The matrix \((F_2 \otimes 1_{N/2})\) is a single Hadamard operation on the most significant qubit. We would like to emphasize that this single quantum operation corresponds to a full butterfly diagram.

The implementation of the twiddle matrix is more complex. Notice that \( T_{N/2} \) can be written as a tensor product of diagonal matrices

\[
T_{N/2} = L_{n-1} \otimes \ldots \otimes L_2 \otimes L_1.
\]

Thus, \( 1_{N/2} \oplus T_{N/2} \) can be realized by controlled phase shift operations. Figure 3 shows the implementation of the two operations discussed so far.

![Figure 3](image)

**Figure 3:** For length \( N=8 \), only three qubits are necessary. The circuit on the left implements \((F_2 \otimes 1_{N/2})\) and the other realizes the twiddle matrix \( 1_4 \oplus \text{diag}(1, \omega, \omega^2, \omega^3) \).

It remains to discuss the other two operations in (1). The operation \((1_2 \otimes F_{N/2})\) means that an implementation of the discrete Fourier transform of length \( N/2 \) is used on the least significant \((n - 1)\) bits. The operation \( \Upsilon_N \) is a permutation of quantum wires. We can combine all the permutations

\[
\Upsilon_N(1_2 \otimes \Upsilon_{N/2}) \ldots (1_{N-2} \otimes \Upsilon_4)
\]

into a single permutation of quantum wires. The resulting permutation is the bit reversal, see Figure 4. The classical and quantum implementation of the discrete Fourier transform of length 8 are compared in Figure 5. We observe that the butterfly diagrams find simple realizations but the twiddle matrices require more elementary quantum gate operations.

The complexity of the quantum implementation can be estimated as follows. If we denote by \( R(N) \) the number of gates necessary to implement the DFT
Figure 4: The bit reversal permutation is given by $1_2 \otimes Y_4$ followed by $Y_8$.

of length $N = 2^n$ on a quantum computer, then equation (1) implies the recurrence relation

$$R(N) = R(N/2) + O(\log N)$$

which leads to the estimate $R(N) = O(\log^2 N)$.

Shor’s factoring algorithm relies on the quantum Fourier transform in a fundamental way. For more details on Fourier transforms and their generalizations to nonabelian groups, see [4, 5].

§4 Quantum Cosine and Sine Transforms

We derive quantum circuits for discrete Cosine and Sine transforms in this section. The main idea is simple: reuse the circuits for the discrete Fourier transform.

The discrete Cosine and Sine transforms are divided into various families. We follow [6] and define the following four versions of discrete Cosine transforms:

$$C_{iN}^I := \left( \frac{2}{N} \right)^{1/2} \left[ k_i \cos \frac{i j \pi}{N} \right]_{i,j=0,...,N}$$

$$C_{iN}^{II} := \left( \frac{2}{N} \right)^{1/2} \left[ k_i \cos \frac{i (j + 1/2) \pi}{N} \right]_{i,j=0,...,N-1}$$

$$C_{iN}^{III} := \left( \frac{2}{N} \right)^{1/2} \left[ k_i \cos \frac{(i + 1/2) j \pi}{N} \right]_{i,j=0,...,N-1}$$

$$C_{iN}^{IV} := \left( \frac{2}{N} \right)^{1/2} \left[ k_i \cos \frac{(i + 1/2) (j + 1/2) \pi}{N} \right]_{i,j=0,...,N-1}$$

where $k_i := 1$ for $i = 1, \ldots, N-1$ and $k_0 := 1/\sqrt{2}$. The numbers $k_i$ ensure that the transforms are orthogonal. The discrete Sine transforms are defined
Figure 5: Dataflow graph of the DFT of length 8 and the corresponding quantum circuits.


\[ S_{N}^{I} := \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} k_i \sin \frac{i j \pi}{N} \end{bmatrix}_{i,j=1,...,N-1} \]

\[ S_{N}^{II} := \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} k_i \sin \frac{i (j + 1/2) \pi}{N} \end{bmatrix}_{i,j=0,...,N-1} \]

\[ S_{N}^{III} := \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} k_i \sin \frac{(i + 1/2) j \pi}{N} \end{bmatrix}_{i,j=0,...,N-1} \]

\[ S_{N}^{IV} := \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} k_i \sin \frac{(i + 1/2)(j + 1/2) \pi}{N} \end{bmatrix}_{i,j=0,...,N-1} \]

where the constants \( k_i \) are defined as above. Notice that \( C_{N}^{III} \) (resp. \( S_{N}^{III} \)) is the transpose of \( C_{N}^{II} \) (resp. \( S_{N}^{II} \)), hence it suffices to derive circuits for the type II transforms.

It is well-known that the trigonometric transforms can be obtained by conjugating the discrete Fourier transform \( F_{2N} \) by certain sparse matrices. We refer the reader to Wickerhauser \[7\] for more details on the decompositions.

**DCT\(_I\) and DST\(_I\).** We derive the circuits for the discrete Sine and Cosine transforms of type I all at once. Indeed, the DST\(_I\) and DCT\(_I\) can be recovered from the DFT by a base change

\[ T_{N}^{I} \cdot F_{2N} \cdot T_{N} = C_{N}^{I} \oplus i S_{N}^{I}, \]

where

\[ T_{N} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & \cdots & \frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 & \cdots & \frac{i}{\sqrt{2}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & \cdots & 1 \\ \frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} & \cdots & \frac{i}{\sqrt{2}} \end{pmatrix} \]

It is straightforward to check that (2) holds, see Theorem 3.10 in \[7\]. Since we already know efficient quantum circuits for the DFT, it remains to find an efficient implementation of the base change matrix \( T_{N} \).

It will be convenient to denote the basis vectors of \( \mathbb{C}^{2^{n+1}} \) by \( |bx\rangle \), where \( b \) is a single bit and \( x \) is an \( n \)-bit number. The two’s complement of an \( n \)-bit unsigned integer \( x \) is denoted by \( x' \), that is, \( x' = 2^{n} - x \). The action of \( T_{N} \)
can be described by

\[ T_N |00\rangle = |00\rangle \quad T_N |0x\rangle = \frac{1}{\sqrt{2}} |0x\rangle + \frac{1}{\sqrt{2}} |1x'\rangle \]
\[ T_N |10\rangle = |10\rangle \quad T_N |1x\rangle = \frac{i}{\sqrt{2}} |0x\rangle - \frac{i}{\sqrt{2}} |1x'\rangle \]

for all integers \( x \) in the range \( 1 \leq x < 2^n \). Ignoring the two’s complement in \( T_N \), we can define an operator \( D \) by

\[ D |00\rangle = |00\rangle \quad D |0x\rangle = \frac{1}{\sqrt{2}} |0x\rangle + \frac{1}{\sqrt{2}} |1x\rangle \]
\[ D |10\rangle = |10\rangle \quad D |1x\rangle = \frac{i}{\sqrt{2}} |0x\rangle - \frac{i}{\sqrt{2}} |1x\rangle \]

for all integers \( x \) in the range \( 1 \leq x < 2^n \). This operator is essentially block diagonal and easy to implement by a single qubit operation, followed by a correction. Indeed, define the matrix \( B \) by

\[ B = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}, \]

then Figure 5 gives an implementation of the operator \( D \).

\[ \begin{array}{c}
|00\rangle \\
|0x\rangle \\
|10\rangle \\
|1x\rangle \\
\end{array} \quad \begin{array}{c}
|00\rangle \\
|0x\rangle \\
|10\rangle \\
|1x\rangle \\
\end{array} \]

\[ \begin{array}{c}
|00\rangle \\
|0x\rangle \\
|10\rangle \\
|1x\rangle \\
\end{array} \quad \begin{array}{c}
|00\rangle \\
|0x\rangle \\
|10\rangle \\
|1x\rangle \\
\end{array} \]

\[ \begin{array}{c}
|00\rangle \\
|0x\rangle \\
|10\rangle \\
|1x\rangle \\
\end{array} \quad \begin{array}{c}
|00\rangle \\
|0x\rangle \\
|10\rangle \\
|1x\rangle \\
\end{array} \]

**Figure 5:** Circuits realizing the block matrix \( D \) and the permutation \( \pi \).

Define \( \pi \) to be the permutation given by a two’s complement conditioned on the most significant bit \( \pi |0x\rangle = |0x\rangle \) and \( \pi |1x\rangle = |1x'\rangle \) for all \( n \)-bit integers \( x \). It is clear that \( T_N = \pi D \). The circuits for the permutation \( \pi \) is shown in Figure 5.

**Theorem 1** The discrete Cosine transform \( C_N^1 \) and the discrete Sine transform \( S_N^1 \) can be realized with at most \( O(\log^2 N) \) elementary quantum gates; the quantum circuit for these transforms is shown in Figure 6.

**Proof.** Let \( N = 2^n \). We note that \( O(\log^2 N) \) quantum gates are sufficient to realize the DFT of length \( 2N \). The permutation \( \pi \) can be implemented with at most \( O(\log^2 N) \) elementary gates. At most \( O(\log N) \) quantum gates are needed to realize the operator \( D \). This shows that the DCT\(_1\) and the DST\(_1\) can be realized with at most \( O(\log^2 N) \) quantum gates. The preceding discussion shows that Figure 6 realizes the DCT\(_1\) and DST\(_1\). \( \square \)
DCT$_{IV}$ and DST$_{IV}$. The trigonometric transforms of type IV are derived from the DFT by

$$e^{\pi i/4N} R_N \cdot F_{2N} \cdot R_N = C^\text{IV}_N \oplus (-i)S^\text{IV}_N.$$  \hfill (3)

Here $R_N$ denotes the matrix

$$R_N = \frac{1}{\sqrt{2}} \begin{pmatrix}
1 & -i & -i\omega & \cdots & -i\omega^{N-1} \\
\omega & -\omega^2 & \omega^3 & \cdots & -\omega^N \\
\omega^2 & -\omega & \omega^2 & \cdots & -\omega^N \\
\omega^3 & -\omega^2 & \omega & \cdots & -\omega^N \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\omega^{N-1} & -\omega & \omega^2 & \cdots & 1
\end{pmatrix}$$

with $\omega$ the primitive $4N$-th root of unity $\omega = \exp(2\pi i/4N)$. Equation (3) is a consequence of Theorem 3.19 in \cite{7} obtained by complex conjugation.

**Theorem 2** The discrete Cosine transform $C^\text{IV}_N$ and the discrete Sine transform $S^\text{IV}_N$ can be realized with at most $O(\log^2 N)$ elementary quantum gates; the quantum circuit for these transforms is shown in Figure 7.
Proof. It remains to show that there exists an efficient quantum circuit for the matrix $R_N$ in equation (3). A factorization of $R_N$ can be obtained as follows. Denote by $\overline{x}$ the one’s complement of an $n$-bit integer $x$. We define a permutation matrix $\pi_1$ by $\pi_1 |0x\rangle = |0x\rangle$ and $\pi_1 |1x\rangle = |1\overline{x}\rangle$ for all integers $x$ in the range of $0 \leq x < 2^n$. Denote by $D_1$ the diagonal matrix

$$D_1 = \text{diag}(1, \omega, \ldots, \omega^{N-1}, \overline{\omega}, \ldots, \overline{\omega^2}, \overline{\omega}).$$

Then $R_N$ can be factored as

$$R_N = \pi_1 \cdot D_1 \cdot \left( \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix} \otimes 1_N \right) = \pi_1 \cdot D_1 \cdot (\overline{B} \otimes 1_N).$$

Note that $\overline{B} \otimes 1_N$ is a single qubit operation, and $\pi_1$ can be realized by controlled not operations. The implementation of the diagonal matrix $D_1$ is more interesting. Note that the diagonal matrices of increasing (decreasing) powers can be written by tensor products

$$\Delta_1 = \text{diag}(1, \omega, \ldots, \omega^{N-1}) = L_n \otimes \cdots \otimes L_2 \otimes L_1$$
$$\Delta_2 = \text{diag}(\overline{\omega}, \ldots, \overline{\omega^{N-1}}, 1) = K_n \otimes \cdots \otimes K_2 \otimes K_1$$

where $L_j = \text{diag}(1, \omega^{2^j-1})$ and $K_j = \text{diag}(\overline{\omega^{2^j-1}}, 1)$. Therefore, it is possible to write $D_1$ in the form $D_1 = (C \otimes 1_N) \cdot (\Delta_1 \oplus \Delta_2)$ with $C = \text{diag}(1, \overline{\omega})$. The circuit for the diagonal matrix $D_1$ is shown in Figure 8.

![Figure 8: Quantum circuit for the diagonal matrix $D_1$.](image)

The complete quantum circuit for the DCT$_{IV}$ is shown in Figure 7. Note that the last three single qubit gates $C, B^\dagger, \text{and } M = \text{diag}(e^{\pi i/4N}, e^{\pi i/4N})$ can be combined into a single gate $MB^\dagger C$. \end{proof}

**DCT$_{II}$ and DST$_{II}$.** The implementation of the trigonometric transforms of type II follows a similar pattern. Both transforms can be recovered from the DFT of length $2N$ after multiplication with certain sparse matrices, cf. Theorem 3.13 in [7]:

...
\[ U_N^\dagger \cdot F_{2N} \cdot V_N = C_N^{II} \oplus (-i)S_N^{II}, \]  
\[ V_N = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & \vdots & 1 \\ 1 & 1 & \vdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \vdots & \vdots & 1 \end{pmatrix}, \]  
and
\[ U_N = \begin{pmatrix} 1 & 0 & \vdots & 0 \\ \omega \frac{N-1}{\sqrt{2}} & -i \frac{\omega^N}{\sqrt{2}} & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \omega \frac{N-1}{\sqrt{2}} & i \omega^{-1} & -1 \end{pmatrix}, \]  
and \( \omega \) denotes the \( 4N \)-th primitive root of unity \( \omega = \exp(2\pi i/4N) \), \( i^2 = -1 \).

**Theorem 3** The discrete Cosine transform \( C_N^{II} \) and the discrete Sine transform \( S_N^{II} \) can be realized with at most \( O(\log^2 N) \) elementary quantum gates; the quantum circuit for these transforms is shown in Figure 9.

**Proof.** We need to derive efficient quantum circuits for the matrices \( V_N \) and \( U_N \) in equation (4). The matrix \( V_N \) has a fairly simple decomposition in terms of quantum circuits.

**Lemma 4** \( V_N = \pi_1(H \otimes 1_N) \).
Proof. It is clear that the Hadamard transform on the most significant bit $H \otimes I_N$ is – up to a permutation of rows – equivalent to $V_N$. The appropriate permutation of rows has been introduced in the previous section, namely $\pi_1 |0x\rangle = |1x\rangle$ and $\pi_1 |1x\rangle = |1\overline{x}\rangle$ for all $0 \leq x < 2^n$. We can conclude that $V_N = \pi_1(H \otimes 1_N)$ as desired. □

The implementation of $U_N$ is more elaborate. Notice that

$$U_N |0\rangle = |0\rangle \quad \quad U_N |x\rangle = \frac{\sqrt{2}}{\sqrt{2}} |0x\rangle + \frac{\omega^x}{\sqrt{2}} |1x\rangle$$

for all integers $x$ in the range $1 \leq x < 2^n$ and all integers $y$ in $0 \leq y < 2^n - 1$. Here $0$ and $1$ denote the $n$-bit integers $0$ and $2^n - 1$ respectively.

Define $D_0$ by $D_0 |0\rangle = |0\rangle$ and $D_0 |x\rangle = |x\rangle$ otherwise. We define a permutation $\pi_2$ by $\pi_2 |0x\rangle = |0x\rangle$ and $\pi_2 |1x\rangle = |1(x+1 \text{ mod } 2^n)\rangle$ for all integers $x$ in $0 \leq x < 2^n$.

Lemma 5 $U_N = D_1^T T_N D_0 \pi_2$.

Proof. Since $D_1^T |0x\rangle = \overline{\omega}^x |0x\rangle$ and $D_1^T |x\rangle = \omega^x |1x\rangle$, we obtain

$$D_1^T T_N |0x\rangle = \frac{\overline{\omega}^x}{\sqrt{2}} |0x\rangle + \frac{\omega^x}{\sqrt{2}} |1x\rangle$$

$$D_1^T T_N |x\rangle = -\frac{\overline{\omega}^x}{\sqrt{2}} |0x\rangle + \frac{i\omega^x}{\sqrt{2}} |1x\rangle$$

We have $D_0 \pi_2 |0x\rangle = |0x\rangle$, $D_0 \pi_2 |1x\rangle = |1(x+1 \text{ mod } 2^n)\rangle$ for all integers $x$ in $0 \leq x < 2^n - 1$, and $D_0 \pi_2 |11\rangle = i |10\rangle$. We note that $(x + 1 \text{ mod } 2^n)' = \overline{x}$, whence combining $D_1^T T_N$ with $D_0 \pi_2$ shows the result. □

Recall that $T_N = \pi D$. It follows that

$$U_N^\dagger = \pi_2^{-1} (\overline{D}_0 D^t) \pi^{-1} D_1.$$ 

The implementation of $D_1$ has been described in the section on the DCT$_{IV}$, and the implementation of $\pi$ (and hence $\pi^{-1}$) is contained in the section on the DCT$_I$. The implementation of $\pi_2^{-1}$ is also straightforward. It remains to find an implementation of $\overline{D}_0 D^t$. We observe that

$$\overline{D}_0 D^t |00\rangle = |00\rangle \quad \quad \overline{D}_0 D^t |0x\rangle = \frac{1}{\sqrt{2}} |0x\rangle + \frac{i}{\sqrt{2}} |1x\rangle$$

$$\overline{D}_0 D^t |10\rangle = (-i) |10\rangle \quad \quad \overline{D}_0 D^t |1x\rangle = \frac{1}{\sqrt{2}} |0x\rangle - \frac{i}{\sqrt{2}} |1x\rangle$$

This can be accomplished by the single bit operation $B^t \otimes 1_N$ followed by a multiply conditioned gate $J = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$. The full circuit is shown in Figure 4. The statement about the complexity is clear. □
§5 Quantum Hartley Transforms

The discrete Hartley transform of length $N \in \mathbb{N}$ is the real $N \times N$ matrix $A_N$ defined by

$$A_N := \frac{1}{\sqrt{N}} \left[ \text{cas} \left( \frac{2\pi ij}{N} \right) \right]_{i,j=0,\ldots,N-1},$$

where the function $\text{cas} : \mathbb{R} \to \mathbb{R}$ is defined by $\text{cas}(x) := \cos(x) + \sin(x)$, see [2, 3] for classical implementations. The property

$$A_N = \left( \frac{1-i}{2} \right) F_N + \left( \frac{1+i}{2} \right) F_N^3$$

is easily seen from the definition. We derive a quantum circuit implementing $A_N$ with one auxiliary quantum bit.

![Circuit realising a quantum Hartley transform](image)

**Figure 10:** Circuit realising a quantum Hartley transform

**Lemma 6** The discrete Hartley transform can be factorized in the form shown in Figure 10. Here $R$ is the unitary circulant matrix

$$R := \frac{1}{2} \left( \begin{array}{cc} 1-i & 1+i \\ 1+i & 1-i \end{array} \right)$$

and $H$ denotes the Hadamard transform.

**Proof.** Let $\tilde{F}_N$ be the transformation which effects a DFT conditioned to the first bit, i.e., written in terms of matrices we have $\tilde{F}_N = I_N \oplus F_N$. We now show that the given circuit computes the linear transformation $|0\rangle |x\rangle \mapsto |0\rangle A_N |x\rangle$ for all unit vectors $x \in \mathbb{C}^n$. Proceeding from left to right in the circuit given in Figure 10, we obtain

$$|0\rangle |x\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |x\rangle$$
\[
\begin{align*}
\tilde{F}_N & \quad \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) F_N |x\rangle \\
\tilde{F}_N^2 & \quad \frac{1}{\sqrt{2}} |0\rangle F_N |x\rangle + \frac{1}{\sqrt{2}} |0\rangle F_N^3 |x\rangle \\
\tilde{R} & \quad \frac{1}{\sqrt{2}} |0\rangle \left( \frac{1}{2} (1 - i) F_N + \frac{1}{2} (1 + i) F_N^3 \right) |x\rangle \\
& \quad + \frac{1}{\sqrt{2}} |1\rangle \left( \frac{1}{2} (1 + i) F_N + \frac{1}{2} (1 - i) F_N^3 \right) |x\rangle \\
& = \frac{1}{\sqrt{2}} |0\rangle A_N |x\rangle + \frac{1}{\sqrt{2}} |1\rangle F_N^{-2} A_N |x\rangle \\
\tilde{F}_N^2 & \quad \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) A_N |x\rangle \\
\tilde{H} & \quad |0\rangle A_N |x\rangle
\end{align*}
\]

as desired. □

**Theorem 7** The discrete Hartley transform \( A_N \) can be computed on a quantum computer using \( O(\log^2 N) \) elementary operations if we allow one additional ancilla qubit.

**Proof.** Recall that the discrete Fourier transform \( F_N \) can be implemented \( O(\log^2 N) \) operations as shown in Section §3. The statement follows from Lemma 6 since all transformations given there require at most \( O(\log^2 N) \) elementary operations. □

§6 Conclusions

We have shown that the discrete Cosine transforms, the discrete Sine transforms, and the discrete Hartley transforms have extremely efficient realizations on a quantum computer. All implementations illustrated an important design principle: the reusability of highly optimized quantum circuits. Apart from a few sparse matrices, we only needed the circuits for the discrete Fourier transform for the implementations. A key point is that an improvement of a basic circuit, like the DFT, immediately leads to more efficient quantum circuits for the DCT, DST, and DHT.

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] Th. Beth. Generating Fast Hartley Transforms - Another Application of the Algebraic Discrete Fourier Transform. In *Proc. URSI-ISSSE '89*, pages 688–692, 1989.

[3] Bracewell. *The Hartley Transform*. Cambridge Univ. Press, 1979.

[4] P. Høyer. Efficient Quantum Transforms. LANL preprint quant-ph/9702028, February 1997.

[5] M. Püschel, M. Rötteler, and Th. Beth. Fast Quantum Fourier Transforms for a Class of non-abelian Groups. In *Proceedings Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-13)*, volume 1719 of *Lecture Notes in Computer Science*, pages 148–159. Springer, 1999.

[6] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, and Applications*. Academic Press, 1990.

[7] V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A.K. Peters, Wellesley, 1993.