The Role of Super-fast Transforms in Speeding up Quantum Computations

Zeljko Zilic and Katarzyna Radecka
McGill University, Montreal, Canada
{kasiar,zeljko}@macs.ece.mcgill.ca

ABSTRACT

We present the role that spectral methods play in the development of the most impressive quantum algorithms, such as the polynomial time number factoring algorithm by Shor. While the fast transform algorithms reduce the number of operations needed in obtaining the transforms from $O(2^{2n})$ to $O(n2^n)$, quantum transforms are in comparison super-fast. The Quantum Fourier Transform can be performed in $O(n^2)$ time, while the specific cases of Walsh-Hadamard and Chrestenson Transforms require only $O(n)$ operations. We derive Quantum Fourier Transform over non-binary quantum digits using the Chrestenson gate, which can serve as a basic block for quantum transforms.

1. INTRODUCTION

There are several reasons for the current keen interest in quantum computing [11],[12],[15],[19],[20]. Quantum computing is capable of speeding up previously hard problems, such as factoring large numbers. It also appears as a natural outcome of the trends in technology scaling. Further, several working quantum-computing systems have been demonstrated [9],[17], and the companies, such as IBM and HP, are starting to invest towards making quantum computing systems, and complementing the previously purely theoretical research in the area.

Quantum computing is perfectly fit for consideration from the point of view of multiple-valued logic (MVL) and spectral methods theory and practice. First, each *qubit*, or binary quantum information unit, actually stores many levels of information, which are manipulated by essentially MVL techniques. Second, some of the circuit realization techniques for quantum logic systems have already been investigated for their suitability in MVL systems [1], [7]. Finally, it appears that the most of the major improvements of quantum computing over the classical computing algorithms are facilitated by the use of orthogonal transforms and the spectral methods.

The invention of classical Fast Fourier Transform (FFT) [5] and related orthogonal transforms is rightly considered to be one of the most important nontrivial algorithms in practice [18]. While fast orthogonal transforms speed up many significant algorithms, the quantum transforms offer even more dramatic improvements.

This paper is aimed at explaining the mechanisms that make the spectral methods extremely useful in quantum computing [14], including the famous Shor's number factoring algorithm [16] that has important cryptographic implication of making the number factorization-based encryption methods breakable. Several classes of quantum transform implementations are presented. Further, we consider the issue of the multiple-valued logic quantum gates. As most quantum computing results have so far been obtained for binary logic, we consider the problem of the design of non-binary quantum logic gates and demonstrate the usefulness of the Chrestenson gate.

The paper is organized as follows. After presenting the basics in Section 2, the common constructions of the quantum gates are presented in Section 3. The role that the orthogonal transforms and spectral techniques play in the fast quantum algorithms are explained in Section 4. In Section 5, we further design the MVL quantum transforms.

2. PRELIMINARIES

According to the quantum mechanics principles, each quantum state is a superposition of multiple *pure states*. Pure states can be observed, as in classical physics computing. This is not true for the quantum states in general. The computing paradigm uses the premises of quantum physics to perform multiple simultaneous computations to outperform the classical computers. A quantum logic system is defined Hilbert space over complex numbers, with well-defined scalar product and the vector norm function. For quantum functions *f* and *g*, defined on *n* elements, a *scalar product* is given by:

$$< f|g >= \frac{1}{\sqrt{n}} \sum_{x=1}^{n} f^*(x)g(x)$$

where the symbol * denotes the complex conjugation. The dot product naturally induces a *norm* as:

$$\|f\| = \sqrt{< f|f >}\,.$$

Then, the *orthonormal basis* of space with *n* quantum basis states is $\{|x_1\rangle, |x_2\rangle, \cdots, |x_n\rangle\}$ if the scalar product of each vector by itself is 1, otherwise the product with any other basis vector is 0. The state of the system is expressed as a linear combination of the basic states, i.e., $a_1|x_1\rangle + a_2|x_2\rangle + \cdots + a_n|x_n\rangle$. The complex-valued amplitudes a_i are referred to as wave functions with respect to the basis $|x_1\rangle, |x_2\rangle, \cdots, |x_n\rangle$.

Proceedings of the 32nd IEEE International Symposium on Multiple-Valued Logic (ISMVL'02)
0195-623X/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

The norm, or the length of the vector, that is obtained as a sum of the squares of amplitudes is always equal to one, i.e. $\sum |a_i|^2 = 1$. The *system evolution* in a quantum system can only be performed through a unitary operation that preserves the norm of the vector of amplitudes. Hence, each system evolution that transforms a state

$$a_1|x_1\rangle + a_2|x_2\rangle + \cdots + a_n|x_n\rangle$$

into a state

$$a'_1|x_1\rangle + a'_2|x_2\rangle + \cdots + a'_n|x_n\rangle$$

can be expressed via a norm-preserving (unitary) mapping U by $\vec{a}' = U\vec{a}$. As a consequence of such a linear dependence, these evolutions are always *reversible*. The definitions of the basic units of information are as follows.

Definition 1 *A quantum bit, or qubit, is a binary quantum system, defined over the Hilbert space H_2 with a fixed basis $\{|0\rangle, |1\rangle\}$. A q-ary quantum digit is a multiple-valued logic system over the Hilbert space H_q with a fixed basis $\{|0\rangle, |1\rangle, \cdots, |q-1\rangle\}$.*

As the binary case is a specific case of q=2, for now we deal with multiple-valued case. The state of a single quantum digit is a vector

$$c_0|0\rangle + c_1|1\rangle + \cdots + c_{q-1}|q-1\rangle$$

where the vector norm is 1:

$$|c_0|^2 + |c_1|^2 + \cdots + |c_{q-1}|^2 = 1$$

The operations that are allowed within the quantum system must preserve the vector norm.

Definition 2*: The operation called an unary quantum gate is an unitary linear mapping from H_{q-1} to H_{q-1}.*

We will study separately binary (q=2) and radix-q (or q-ary) gates. All such gates are represented by unitary matrices over complex numbers. The standard complex vector and matrix notation is summarized in Table 1.

x^*	Complex conjugate of complex number x: $(a+ib)^* = a - ib$
A^*	Complex conjugate of (all entries of) A
A^T	Transpose of matrix A
$A^{\mp}$	Hermitian conjugate (adjoint) of matrix A: $\begin{bmatrix} a & c \\ b & d \end{bmatrix}^{\mp} = \begin{bmatrix} a^* & b^* \\ c^* & d^* \end{bmatrix}$
$\langle a \mid b \rangle$	Scalar product of vectors $\|a\rangle, \|b\rangle$
$\|a\rangle \otimes \|b\rangle$ or $\|a\rangle\|b\rangle$	Tensor product of vectors $\|a\rangle, \|b\rangle$

Table 1: Standard Quantum Operation Notation

The matrices of the mappings in quantum systems are unitary. For unitary matrix A, we have: $AA^{\mp} = I$ **.**

2.1 Combining the States – Entanglement

According to quantum mechanics, the combination of quantum state digits can be in either *decomposable* or in *entangled* states. Each individual state digit can be observed in the former case, but not in the later. The state space can be compounded from several states by considering their Cartesian product. In terms of the quantum state functions, the compounded state is expressed as

$$\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} |x_i, y_j\rangle .$$

We say that such a state is decomposable if it can be represented as

$$\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} |x_i, y_j\rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} a_i b_j |x_i\rangle |y_j\rangle =$$

$$= \sum_{i=1}^{n} a_i |x_i\rangle \sum_{j=1}^{m} b_i |y_i\rangle .$$

Otherwise, the state is entangled.

Example 1: a) Consider a system of two qubits, given as

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) .$$

This system is decomposable, as the actions of the first and the second qubit are disentangled.

b) The system $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ is entangled, as no decomposition is possible, which is easily seen by inability to satisfy the set of equations describing the same state in the decomposed form. ♦

In general, the states are compounded by means of the tensor (Kronecker) product of the basic state spaces. Such a combination is referred to as the *quantum register*. It appears that the speedups in quantum computation are due to the entanglement, by which many computations are performed in parallel. In that sense, entanglement is a special new resource in quantum computing.

3. QUANTUM GATES

Quantum operations require the existence of several types of quantum gates that act as unitary mappings over Hilbert space. The quantum gates are always reversible (but not all reversible gates are quantum). Since all transformation matrices are unitary, it follows that their inverses are equal to their Hermitian conjugates.

IEEE COMPUTER SOCIETY

3.1 Binary Quantum Gates

Few gates that are useful in developing quantum transforms are given next. For reasons of reversibility, each gate must have the same number of inputs and the outputs. The single input gates are defined by a 2x2 unitary matrix with possibly complex entries.

Example 2: Consider the quantum gate defined as:

$$W_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

The gate is called Walsh, or Walsh-Hadamard gate. The actions performed over the basis vectors are:

$$W_2|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \quad W_2|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).$$

This gate is a square root of the identity gate, since $(W_2)^2 = I$. It is one of the most useful quantum gates. ♦

The gates with n inputs and n outputs are given by a unitary matrix with 2^n x 2^n entries. Multiple-input, multiple output gates performe some unitary operation over a tensor product of single quantum Hilbert spaces. There are several common constructions of multiple qubit gates [15].

Example 3: The controlled NOT, or CNOT gate accepts two quantum inputs, a and b, and produces the outputs: a and $a \oplus b$. Its transform matrix in input order a, b is:

$$N_C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \qquad \blacklozenge$$

The *controlled gate* approach is a generic multi-qubit construction applied to any single-qubit gate. In the controlled gate construction, the first (control) qubit is left unchanged, while the second one is affected by an unitary transformation if the first qubit is in the state |1>. It can be verified that the gate in Example 3 inverts the second bit only if the first bit is 1, otherwise both bits stay unchanged.

Example 4: The phase shift gate S_a performs the following multiplication (some authors refer to it as "scaling"):

$$S_a = \begin{bmatrix} e^{ia} & 0 \\ 0 & e^{ia} \end{bmatrix}.$$

The controlled phase shift gate performs the operation if the value of the control qubit is 1. The symbol for the controlled shift gate R_a is shown in Figure 1. The effect of the circuit is summarized as:

$|00\rangle \to |00\rangle, |01\rangle \to |01\rangle, |10\rangle \to e^{ia}|10\rangle, |11\rangle \to e^{ia}|11\rangle$. ♦

Other multi-qubit constructions include that of n independent single qubit operations. In that case, the transform matrix is the Kronecker product of the single-qubit gate matrices. Additionally, *swaps*, or, in general, permutations of multiple inputs present another way of constructing multiple qubit gates.

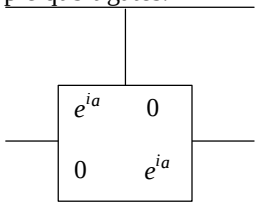

Figure 1: Controlled Phase Shift Gate

3.2 Q-ary Quantum Gates

There are no reasons to assume that quantum physics prefers binary quantum states. Quantum computing can hence rely on the construction of gates in the q-valued case, over the space H_q. In general, any unary transformation on H_q presents a valid q-ary quantum gate. However, we know of few explicit constructions.

For our case, we found that the generalization of the Walsh-Hadamard gate is very useful. We call it Chrestenson gate, as it performs the basic step of the Chrestenson transform [13].

Example 5: Consider the case q=3. The complex 3rd root of unity is:

$$a = e^{-\frac{2pi}{3}} = \cos(\frac{-2}{3}pi) + i\sin(\frac{-2}{3}pi) = -0.5 - i*0.866.$$

Using the root of unity, the Chrestenson gate performs the mapping in H_3 given by the following matrix:

$$CH = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}.$$

It is easy to check that the gate is unitary by confirming that the product with its own Hermitian conjugate is equal to the identity matrix. The conjugate is of form:

$$CH^{\mp} = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}^{\mp} = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a^* & \left(a^2\right)^* \\ 1 & \left(a^2\right)^* & a^* \end{bmatrix}.$$

The third roots of the unity satisfy the properties $a^* = a^2$ *and* $(a^2)^* = a$, leading to the product

$$CH\ CH^{\mp} = \frac{1}{3}\begin{bmatrix} 3 & 1+a+a^2 & 1+a+a^2 \\ 1+a+a^2 & 1+a^3+a^3 & 1+a+a^2 \\ 1+a+a^2 & 1+a+a^2 & 1+a^3+a^3 \end{bmatrix}$$

After applying identities $a^3 = 1$ *and* $1+a+a^2 = 0$ we obtain the identity matrix I. ♦

0195-623X/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

Many more 3-ary qubit gates could be used in quantum circuits over non-binary digits. We point to the construction of q-ary gates from binary quantum gates in [15].

4. QUANTUM TRANSFORMS

The Fourier Transform in general represents elements over an arbitrary Abelian (commutative) group by an expansion along orthogonal set of basis vectors. While even in the most abstract settings, all useful properties that we expect from Fourier transforms hold, we restrict for our purposes the consideration to the (products of) additive groups Z_2 and Z_q, where the group operation is the addition modulo 2 and q, respectively. These correspond to the multivariate binary and multiple-valued transforms, as well as to the classical univariate DFT.

4.1 Transform Basics

The transform is defined as an expansion with respect to the basis functions that are *orthonormal*. For each pair of basis functions, the dot product is 0, unless the two functions are equal, in which case it is 1. In binary case, such an orthogonal basis can be obtained by expanding the Fourier basis to the multivariate binary inputs in the following way. For input variables $x_0, x_1 \ldots, x_{n-1}$, each subset S of the variables is associated with a basis function:

$$c_{\{S\}} = \exp\left(pi \sum_{l \in S} x_l \right) = \begin{cases} +1 if \ \sum l \in S \ is \ even \\ -1 if \ \sum l \in S \ is \ odd \end{cases}.$$

Boolean functions are represented by basis functions as:

$$W(f) = \sum_{\{S\}} c_{\{S\}} c_{\{S\}}.$$

Each *spectral coefficient* $c_{\{S\}}$ is a projection to its basis vector, obtained by the dot product $c_{\{S\}} =< f | c_{\{S\}} >$. Alternatively, spectral coefficient $c_{\{S\}}$ is also equal to the function correlation with the XOR function: $\oplus_{l \in S} x_l$.

We now concentrate on the main underlying efficient quantum algorithm, that for computing Quantum Fourier Transform (QFT). The QFT is defined in the same way as the ordinary Fourier Transform, although the orthogonal basis is the one used for a quantum system. In binary case, the corresponding transform is the quantum Walsh-Hadamard transform, while otherwise we talk about the Chrestenson transform. The former is defined as the Kronecker product of the transform matrix W_2 from *Example 2*. In the binary case, the expansion is of the form

$$H_n |x\rangle = \sum_{y \in \{0,1\}^n} (-1)^{xy} |y\rangle$$

The Chrestenson transform is defined by the expansion that involves complex q-th roots of unity

$$CH_n |x\rangle = \frac{1}{\sqrt{q^n}} \sum_{y \in \{0,1,\cdots,q-1\}^n} e^{-\frac{2pixy}{q}} |y\rangle$$

as well as by means of the Kronecker product of the single-dimensional transform matrix.

For practical calculations of the Chrestenson transform, the input and output vectors are expressed in terms of its q-ary expansions:

$$x = \sum_{i=1}^{n} x_i q^i, y = \sum_{i=1}^{n} y_i q^i$$

and the transform matrix entries have the form:

$$CH_n(x, y) = \frac{1}{\sqrt{q^n}} \sum_{y \in \{0,1,\cdots,q-1\}^n} e^{-\frac{2pixy}{q}} \sum_{i=1}^{n} x_i y_i$$

For q=3, the complex root of unity is $a = e^{-\frac{2pi}{3}}$, and the corresponding transform matrix has the recursive form

$$CH_n = \begin{bmatrix} CH_{n-1} & CH_{n-1} & CH_{n-1} \\ CH_{n-1} & aCH_{n-1} & a^2 CH_{n-1} \\ CH_{n-1} & a^2 CH_{n-1} & aCH_{n-1} \end{bmatrix}$$

4.2 Superfast Quantum Transforms

We now present the derivation of the efficient quantum Fourier transforms. While calculationss are similar to the recursive development of the Fast Fourier Transform, the decomposable quantum states are the main contributor to the algorithm speedups [6]. When dealing with higher dimensionality, we denote the Kronecker product of the quantum basis functions in the shorthand form, i.e., as the product of the basis functions.

4.2.1 Quantum Walsh-Hadamard Transform

The first construction is given in terms of the radix-2 transform. In the case of the transform over Z_2^n, i.e., of n-variable quantum functions over binary digits, the Fourier transform equals the quantum Walsh-Hadamard Transform, which is defined in the analogous way to the ordinary Walsh-Hadamard transform.

$$H_n |x\rangle = \sum_{y \in \{0,1\}^n} (-1)^{xy} |y\rangle$$

for which we use the shorthand notation:

$$|x\rangle \to \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{xy} |y\rangle.$$

IEEE COMPUTER SOCIETY

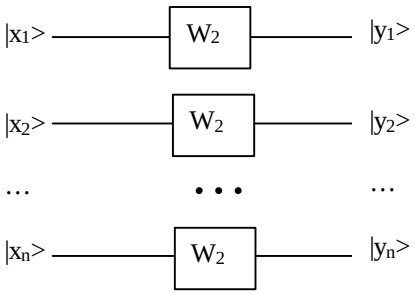

Figure 2: Circuit for Quantum Walsh-Hadamard Transform

The transformation of *n*-variable functions is performed by Kronecker product of univariate transforms. Since the *n*-fold Kronecker product of functions performed over a single quantum digit is equal to the parallel application of these *n* single-qubit functions, the overall transform is performed by only *n* Walsh-Hadamard gates, as in Figure 2. Compared to the classical case, with $O(n2^n)$ operations for the fast transform, the quantum transform is hence super-fast. It is exhibiting the exponential speedup, by requiring only $O(n)$ operations, all of which can be performed in parallel.

Walsh-Hadamard Transform is critical to several key quantum computing transformations. A very common and useful construction, called "Hadamard twice", first transforms a standard basis vector, into a dual basis. The operations are then performed in the dual bases, and the vector is converted back.

4.2.2 Quantum Chrestenson Transform

The generalization of the transform to the *q*-ary case has been traditionally known as the Chrestenson Transform. The transform is defined over $Z_2{}^n$ as:

$$CH_n|x\rangle = \frac{1}{\sqrt{q^n}} \sum_{y\in\{0,1,\cdots,q-1\}^n} e^{\frac{-2pixy}{q}} |y\rangle$$

The multipliers are equal to q^{th} complex roots of the unity. The shorthand notation again denotes that the mapping is performed from the standard basis to the transform basis:

$$|x\rangle \to \frac{1}{\sqrt{q^n}} \sum_{y\in\{0,1,\cdots,q-1\}^n} e^{\frac{-2pixy}{q}} |y\rangle$$

Analogous to the previous case, the super-fast quantum transform is performed by *n* Chrestenson gates in parallel. Chrestenson Transform plays the role equivalent to that of the Walsh-Hadamard Transform when the quantum system is defined over non-binary quantum digits.

4.3 Quantum Discrete Fourier Transform

In general case, quantum transforms defined over an arbitrary set of values will become somewhat more complex to perform. We are mostly interested in transforms over discrete sets. The transform has the univariate form:

$$|x\rangle \to \frac{1}{\sqrt{q}} \sum_{y\in\{0,1,\cdots,q-1\}} e^{\frac{-2pixy}{q}} |y\rangle \quad .$$

Unlike previous cases, there is no apparent general way to represent the transform by means of the Kronecker product of the decomposable bases. Similar to the developments in the traditional Fast Discrete Fourier Transform, several factorizations are possible. One such factorization uses the Chinese Remainder Theorem to express the transform over a set with *m* elements in terms of its remainders modulo the divisors of *m*.

4.3.1 Transforming 2^n Elements - Qubit Description

An alternative factorization approach is much more uniform. For this, we represent up to 2^n input quantities by its *n* qubit representation. In binary case, the inputs and outputs are represented as $x = x_0 + 2x_1 + 2^2 x_2 + \cdots 2^{n-1} x_{n-1}$ and the decomposable quantum representation is: $|x\rangle = |x_0\rangle|x_1\rangle|x_2\rangle\cdots|x_{n-1}\rangle$.

Most of the major achievements in deriving fast quantum algorithms are due to the way to realize the decomposable transform over such a system. We rewrite the transform definition, accounting for the encoding by qubits. The decomposition is fairly analogous to the classical FFT.

Application of the Quantum Fourier Transform to the vector $|y\rangle = |y_0\rangle|y_1\rangle|y_2\rangle\cdots|y_{n-1}\rangle$ is performed as:

$$\frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} e^{-\frac{2pi}{2^n}xy} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} e^{-\frac{2pi}{2^n}x\sum_{l=1}^{n} y_l 2^l} |y_0 y_1 \cdots y_{n-1}\rangle$$

The key step, analogous to the FFT, is the recursive decomposition by the least significant bit:

$$\sum_{y\in\{0,1\}^n} e^{-\frac{2pi}{2^n}xy} |y\rangle = \sum_{y'\in\{0,1\}^{n-1}} e^{-\frac{2pi}{2^n}x2y'} |y'0\rangle + \sum_{y'\in\{0,1\}^{n-1}} e^{-\frac{2pi}{2^n}x(2y'+1)} |y'1\rangle$$

that is algebraically reduced to the decomposable states:

$$\sum_{y'\in\{0,1\}^{n-1}} e^{-\frac{2pi}{2^n}x2y'} |y'\rangle \left(|0\rangle + e^{-\frac{2pix}{2^n}} |1\rangle \right),$$

By extending the same construction recursively, the overall QFT is reduced to (factor $1/\sqrt{2^n}$ omitted):

$$\left(|0\rangle + e^{\frac{-pix}{2^0}} |1\rangle \right) \left(|0\rangle + e^{\frac{-pix}{2^1}} |1\rangle \right) \cdots \left(|0\rangle + e^{\frac{-pix}{2^{n-1}}} |1\rangle \right). \qquad (1)$$

The final step in deriving the efficient transform uses properties of the binary encoding of *x*. Considering the term with 2^l in the denominator, we expand $\exp(2\pi i x/2^l)$:

IEEE COMPUTER SOCIETY

$$\exp\left(\frac{2pi\left(2^{n-1}x_{n-1}+2^{n-1}x_{n-1}+\cdots+2x_1+x_0\right)}{2^l}\right)=$$

$$=1*1\cdots1*\exp(\frac{pix_{l-1}}{2^0})\exp(\frac{pix_{l-2}}{2^1})\cdots\exp(\frac{pix_1}{2^{l-2}})\exp(\frac{pix_0}{2^{l-1}})=$$

$$=(-1)^{x_{l-1}}\exp(\frac{pix_{l-2}}{2})\cdots\exp(\frac{pix_1}{2^{l-2}})\exp(\frac{pix_0}{2^{l-1}}).$$

This expansion is rewritten using the fractional number obtained by dividing x with 2^l, resulting with:

$$\left(|0\rangle+e^{-2pi0.x_0}|1\rangle\right)\left(|0\rangle+e^{-2pi0.x_1x_0}|1\rangle\right)\cdots\left(|0\rangle+e^{-2pi0.x_{n-1}x_{n-2}\cdots x_0}|1\rangle\right)$$

where $0.x_{l-1}x_{l-2}\ldots x_0$ denotes the fraction obtained from the least significant l qubits of x as:

$$0.x_{l-1}x_{l-2}\cdots x_0=2^{-1}x_{l-1}+2^{-2}x_{l-2}+\cdots+2^{-l}x_0.$$

4.3.2 Implementation by Quantum Gates

The last, product representation is amenable to the efficient quantum circuit implementations. It is easy to verify that each term, multiplied with appropriate constant, is the unitary transform. The quantum circuit can be built using the primitives such as Walsh-Hadamard gate and the circuit derived from the rotation gate.

We now show that each term in (1) can be obtained by one or more unitary gates. First, note that a division by the square root of two can be associated with each bracket in (1), making it the unitary operation by itself.

The leftmost product term that is not equal to 1 is implemented by the Walsh-Hadamard gate, as

$$\frac{1}{\sqrt{2}}\left(|0\rangle+e^{-pix}|1\rangle\right)=\frac{1}{\sqrt{2}}\left(|0\rangle+(-1)^{x_0}|1\rangle\right)$$

which is, by the definition, equal to the function performed by the Walsh-Hadamard gate.

The same procedure is applied to the remaining terms. Considering the l^{th} qubit, we first apply the Walsh-Hadamard Transform, leading to the state:

$$\frac{1}{\sqrt{2}}\left(|0\rangle+(-1)^{x_{l-1}}|1\rangle\right)$$

that is further manipulated by a series of unitary transformations. For each qubit x_k, where k is less than l, a phase shift by the operator

$$\exp(\frac{-pix_k}{2^{l-k}})$$

is performed, conditional on k^{th} qubit being nonzero. Recalling Example 4, this operation is performed by the controlled phase shift gate R_k. For each such qubit, one phase shift gate is employed, leading to l-1 gates in total.

The overall quantum transform can hence be implemented by a quantum circuit employing the total of the $n*(n+1)/2$ gates, of which n gates are Walsh-Hadamard, as shown in Figure 3.

4.4 Applications of Quantum Transforms

Among the first problem in which quantum computing was shown to improve searches was Deutsch's problem of guessing if a Boolean function was constant or balanced [8]. Surprisingly, it was shown that by applying "Hadamard twice", only one function evaluation is needed.

Starting with the breakthrough algorithm by Shor [16], a class of algorithms was discovered to solve important problems with cryptographic applications. The algorithm uses QFT to find the periodicity of a number with respect to a random smaller number, resulting in the randomized factoring algorithm. A similar algorithm was developed in [16] for the discrete logarithm.

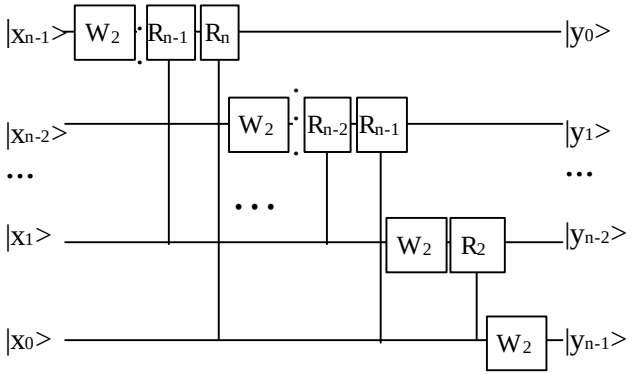

Figure 3: Circuit for QFT by Binary Quantum Gates

Another significant quantum algorithm was developed in 1996 by Grover [10]. The algorithm speeds up searches for hard-to-find, easy-to-verify solutions, such as for NP-complete problems. The algorithm uses Walsh-Hadamard Transform for equally weighted superposition of pure states and to amplify the probability of finding a solution.

5. QFT USING TERNARY QUANTUM GATES

We now present the derivation of Quantum Fourier Transform over 3^n elements, using the ternary quantum circuits. The transform is fairly analogous to the one shown in the binary case, but for brevity, we show it performed as:

$$H_n|x\rangle=\frac{1}{\sqrt{3^n}}\sum_{y\in\{0,1,2\}^n}e^{\frac{-2pixy}{3^n}}|y\rangle=$$

$$=\frac{1}{\sqrt{3^n}}\sum_{y\in\{0,1,2\}^n}e^{-\frac{2pi}{3^n}x\sum_{l=1}^{n}y_l3^l}|y_0y_1\cdots y_{n-1}\rangle=$$

$$=\frac{1}{\sqrt{3^n}}\otimes_{l=0}^{n-1}\left[|0\rangle+e^{-2pix3^{-l}}|1\rangle+e^{-4pix3^{-l}}|2\rangle\right]=$$

$$=\frac{1}{\sqrt{3^n}}\left(|0\rangle+e^{-2pi0.x_0}|1\rangle+e^{-4pi0.x_0}|2\rangle\right)\cdots$$

$$\cdots\left(|0\rangle+e^{-2pi0.x_{n-1}\cdots x_1x_0}|1\rangle+e^{-4pi0.x_{n-1}\cdots x_1x_0}|2\rangle\right)$$

0195-623X/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

We use in this case the ternary fraction notation:

$$0.x_{l-1}x_{l-2}\cdots x_0 = 3^{-1}x_{l-1} + 3^{-2}x_{l-2} + \cdots + 3^{-l}x_0$$

in the final form. The actual circuit for computing the transform is derived by realizing each bracket in the expression with ternary quantum gates.

Starting with the leftmost bracket, the expression

$$T(x_0) = \frac{1}{\sqrt{3}}\left(|0\rangle + e^{-2pi0.x_0}|1\rangle + e^{-4pi0.x_0}|2\rangle\right)$$

can be realized by a single linear and unitary gate. By plugging in all three possible values for x_0 we obtain exactly the action of the Chrestenson gate, as in Sec. 3.2.

The overall circuit is constructed by realizing each bracket. As in the binary case, applied to l^{th} bit is the ternary phase shift operation controlled by x_k $(k<l)$:

$$\exp(\frac{-pix_k}{3^{l-k}})$$

Please note that these controlled shifts involve the digits in the fractional expansion of x, which are beyond the most significant digit.

After putting the pieces together, there is an implementation consisting of one Chrestenson gate per quantum digit, followed by the controlled phase shift gates.

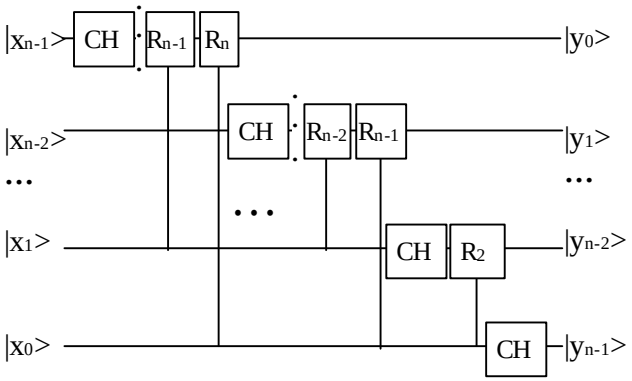

Figure 4: Circuit QFT by Q-ary Quantum Gates

We conclude that the fast Quantum Fourier Transform implementation over ternary digits is performed in the way analogous to the binary case, as shown in Figure 4.

6. CONCLUSIONS AND FUTURE WORK

We reviewed the role of the quantum transforms in the design of efficient quantum algorithms. The circuits for fast quantum transforms are derived for example transforms over several domains.

More work can be done in the design of quantum transform circuits over binary and ternary quantum gates. The implementations that are optimized for the circuit depth have recently been proposed in [4]. It is worth investigating if the same speedup techniques are applicable to ternary quantum circuit constructions. Finally, as the quantum machines are being physically implemented to verify the Shor and Grover algorithms, and as some of currently promising technologies might become real, it will be worth reconsidering the quantum transform circuits with such implementations in mind.

References

[1] Anas Al-Rabadi "Synthesis and Canonical Representations of Equally Input-Output Binary and Multiple-Valued Galois Quantum Logic",*Technical Report #2001/007*, ECE Department, Portland State University, 22nd August 2001

[2] T. Baba, "Development of Quantum Functional Devices for Multiple-Valued Logic Circuits", *Proceedings of International Symposium on Multiple-Valued Logic*, pp. 2-9, 1999.

[3] I. L. Chuang, N. Gershenfeld and M. Kubinec, "Experimental Implementation of Fast Quantum Searching", *Physical Review Letters*, 80(15), 3408-3411, 1998.

[4] R. Cleve and J. Watrous, "Fast Parallel Circuits for the Quantum Fourier Transform", *Proceedings of IEEE Symposium on the Theory of Computing*, pp. 526-535, 2000.

[5] W. Cooley and J. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation*, 19:297-301, 1965.

[6] D. Coppersmith, "An Approximate Fourier Transform Useful in Quantum Factoring", *IBM Research Report RC 19642*, 1994.

[7] X. Deng, T. Hanyu and M. Kameyama, "Quantum Device Model Based Super Pass Gate for Multiple-Valued Digital Systems", *Proceedings of IEEE International Symposium on Multiple-Valued Logic*, pp. 130-138, 1995.

[8] D. Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer", *Proc. Royal Society London A* 400:97-117, 1985.

[9] N. Gershenfeld and I. L. Chuang, "Bulk Spin-Resonance Quantum Computing", Nature, vol. 404, pp. 350-356, 1997.

[10] L. Grover, "A Fast Quantum-Mechanical Algorithm for Database Search", *Proc. 28th ACM Symposium on Theory of Computation*, pp. 212-219, 1996.

[11] J. Gruska, "*Quantum Computing*", McGraw-Hill, 1999.

[12] M. Hirvensalo, "*Quantum Computing*", Springer, 2001.

[13] S. L. Hurst, D. M. Miller and J. Muzio, "Spectral Techniques in Digital Logic", Academic Press, London, 1985.

[14] R. Josza, "Quantum Algorithms and the Fourier Transform", *Proceedings of Royal Society of London*, 454:323-337, 1997.

[15] M. L. Nielsen and I. L. Chuang, "*Quantum Computation and Quantum Information*", Cambridge University Press, 2000.

[16] P. W. Shor, "Polynomial Time Algorithms for Prime Factorization and Discrete Logarithm", *SIAM Journal of Computing*, 26(5): 1484-1509, 1997.

[17] M. Steffen, L. M. K. Vandersypen and I. L. Chuang, "Toward Quantum Computation: A Five-Qubit Quantum Processor", *IEEE Micro*, Vol. 27, No. 1, pp. 24-34, 2001.

[18] J. von zur Gathen and J. Gerhard, "Modern Computer Algebra", Cambridge University Press, 1999.

[19] C. P. Williams and S. H. Clearwater, "*Explorations in Quantum Computing*", Springer Verlag, New York, 1998.

[20] C. P. Williams and S. H. Clearwater, "*Ultimate Zero and One – Computing at the Quantum Frontier*", Springer, 2000.

IEEE COMPUTER SOCIETY