A simplification of the Shor quantum factorization algorithm employing a quantum Hadamard transform

Rupert C. D. Young, Philip M. Birch, Chris R. Chatwin

Department of Engineering and Design,
School of Engineering and Informatics
University of Sussex,
Brighton, United Kingdom, BN1 9QT
E-mail: r.c.d.young@sussex.ac.uk

ABSTRACT

The Shor quantum factorization algorithm allows the factorization or large integers in logarithmic squared time whereas classical algorithms require an exponential time increase with the bit length of the number to be factored. The hardware implementation of the Shor algorithm would thus allow the factorization of the very large integers employed by commercial encryption methods. We propose some modifications of the algorithm by employing some simplification to the stage employing the quantum Fourier transform. The quantum Hadamard transform may be used to replace the quantum Fourier transform in certain cases. This would reduce the hardware complexity of implementation since phase rotation gates with only two states of 0 and π would be required.

Keywords: Shor factorization algorithm, quantum Fourier transform, quantum Hadamard transform, quantum computing

1. INTRODUCTION

Previously we have compared the central importance of the Fourier transform in classical coherent optical processing systems and in algorithms proposed for implementation in quantum computers noting the similarities, and fundamental differences, between the classical and quantum Fourier transform[1,2]. In this paper, we consider in more detail the Shor quantum factorization algorithm from a signal processing viewpoint, emphasizing the importance of the quantum Fourier transform (QFT) in its operation. Approximations to the QFT will be considered with the limit being the quantum Hadamard transform, requiring only binary phase control of the wavefunction rather than the fine phase control implied by an exact QFT implementation. However, in many examples of period finding, non-periodic matching leads to spectral leakage effects which will disrupt operation of the quantum Hadamard transform, although it may be possible to improve this with a window function.

2. THE SHOR ALGORITHM FOR LARGE INTEGER FACTORIZATION

The factorization of large integer numbers into their prime factors is a difficult problem, computations scaling heavily with *n*, the bit length of the integer. The ease of forming a large number as the product of two smaller factors, but in contrast the considerable difficulty in determining these from the composite number if they are unknown, forms the basis of commercial public key encryption systems such as RSA[3]. Thus efficient factorization algorithms would allow defeating such encryption schemes and so be of great technical importance, as well as having more theoretical

applications to discovering new large prime numbers in pure mathematics.

The most efficient classical algorithm is the general number field sieve which scales, for an n-bit number as approximately $O(\exp((\log n)^{1/3}(\log\log n)^{2/3}))$. In contrast, the Shor algorithm[4], by employing the parallelism inherent to quantum computation, scales as $O(n^2(\log n(\log\log n))$, i.e. exponentially faster than the best known classical algorithm. However, the physical implementation of a quantum computer capable of implementing the Shor algorithm has proved very difficult due to the necessity to maintain the required superposition states through the many quantum logic gates required to implement the algorithm for realistically useful integer sizes. To date, the largest number factorized using a quantum computer implementing Shor's algorithm is the number 21, which employed optical techniques for its hardware realization.[5] Recently, proposals have been made for the scaling of integrated ion trap based quantum logic gates and the resources estimated to scale such a device to factor the long bit length numbers employed in cryptographic systems.[6]

The Shor algorithm[4,7] factorises an integer N by finding the period, r, of a function $f(a) = x^a \bmod N$, where x must be in the range $1<x<N$ and co-prime with N. The period r is known as the order and must be an even number for the method to work. Thus if a choice a x results in an odd order, x must be reselected and the $f(a)$ recalculated. Given a suitable r is found, factors of N can be determined by first calculating $p_1 = x^{r/2} - 1$ and $p_2 = x^{r/2} + 1$. By then calculating the greatest common divisor of p_1 and then p_2 with N, i.e. $\gcd(p_{1,2}, N)$, two factors of N can be determined. For large numbers, the gcd can be calculated efficiently using the method of continued fractions[7].

However, employing conventional computing methods requires exponential time in N to determine the factors. The Shor algorithm proposes generating a series of superposition states of the modular functions followed by the Fourier decomposition of the state by a quantum Fourier transform (QFT) operation. Some of these states will comprise a spatially periodic wavefunction, of period r, generated by a successful application of the modular arithmetic procedure for a good choice of x. The unitary transformation effected by the QFT will not result in the collapse of this wavefunction but a concentration of the probability distribution in the location of the detector cell corresponding the value of r. Repeated measurements on successive experiments will then result in a high probability that a detection at this location will be made, enabling the periodicity of the wavefunction r to be determined and hence two factors of N by determining the greatest common divisors of p_1 and p_2 with N [4,7].

The algorithm requires entanglement of bits in a quantum register and unitary operations on these data that perform parallel computations of the superposition state of the data without collapse of the wavefunction. In order to factor a number N, a number q is chosen such that $N^2<q<2N^2$ to create a state in an input quantum register:

$$\left|\psi_1\right\rangle = \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\left|a,0\right\rangle \tag{1}$$

from which is computed $x^a \bmod N$ for all a values from 0 to q-1 which is accomplished in parallel by the quantum computer. This generates the periodic function whose period, r, can be used to determine a factor of N by the classical procedure outlined above. The results are stored in an output register which is entangled with the first register to produce the entangled state:

$$\left|\psi_2\right\rangle = \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\left|a, x^a \bmod N\right\rangle \tag{2}$$

Next, a measurement is made on the output register. This will collapse to any of the values in the output register. However, since the output register is entangled with the first register, the first register will partially collapse to a state consistent with this measurement which results in a superposition state of the first register which is a periodic impulse-like *comb* function that can be related to the order, r, from which a factor of N can be derived[4,7]. However, the period cannot be extracted directly from equation (2) as there is an unknown offset in the *comb* function, say d, as it is not known to which value of $x^a \bmod N$ (in any one period of this function) the output register collapsed. The solution devised by Shor was to perform a QFT on the superposition state of the first register after the measurement of the second register causes this partial collapse to some unknown value of a, say m. This may be written:

$$|\psi_3\rangle = \frac{1}{\sqrt{q}} \sum_{m=0}^{q-1} \sum_{a=0}^{q-1} e^{-j\pi am/q} |m, x^a \bmod N\rangle \tag{3}$$

This will lead to an impulse-like train with a spacing of 1/*r* by a standard result for the discrete Fourier transform[9]. In addition, and most importantly, the impulse train will be centred on-axis, the unknown translation, *d*, of the original impulse train (of periodicity *r*) having been transformed to a linear phase factor. Since the probability of collapse of the wavefunction is proportional to the modulus squared of its values, this will not affect measurements of the impulse train periodicity 1/*r*. The high probability of collapse at the location of the peak of the wavefunction will allow the frequency of the impulse train to be determined, and hence the reciprocal value *r*, thus in turn allowing a factor of *N* to be found.

This process can perhaps be made clearer by an explicit, simple numerical example. The factorization of the number 15 in to 3 an 5 is often used as example as 15 is the smallest number with two prime factors[13]. We take, for example, a seed value $x = 7$ and calculate $7^a \bmod 15$ to yield: 1,7,4,13,1,7,4,13…. We see this results in a periodic function of period (order) equal to 4. We divide by 2 and raise the seed to this power i.e. $7^{4/2}$ to give 49 (note that this only works when the order is even; if it is odd we start again with a new trial seed value). Adding and subtracting 1 gives 48 and 50. The gcd of 15 and 48 is 3 and the gcd of 50 is 5, giving us 3 and 5 as the factors of 15. Thus if we can determine the order of the function generated by the modular exponentiation, we can factorize *N*. The Shor algorithm for determining the periodicity *r* using this example is summarised in graphical form in Figure 1 below. For large *N* calculating the modular exponentiation has to be repeated more than N^2 times which is not feasible on a conventional computer. However, if all values can be computed in parallel via a quantum computation, the modular exponentiation can be carried out in $O(n^3)$ for a *n*-bit number *N*. The period can then be determined from this using the QFT and from this the factors of *N*. However, to achieve this in *O*(poly log *n*) time rather than the exponential time required for a classical implementation requires the entanglement of two quantum registers and a subsequent unitary operation, i..e. a QFT, on the partially collapsed state of one of the registers[4]. However, since the modular exponentiation requires $O(n^3)$ and the QFT (n^2) quantum gates for implementation, the implementation difficulties of the quantum algorithm are, as has been noted above, very technically demanding.

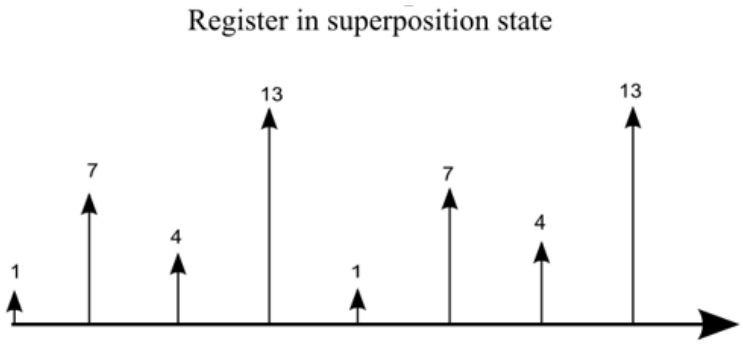

Figure 1 (a). Diagrammatic summary of periodicity determination with the Shor algorithm. A sample is shown of the periodic function resulting from the factorization of 15 with seed $x = 7$, resulting in a periodic function of order 4. Register sample shown before a measurement.

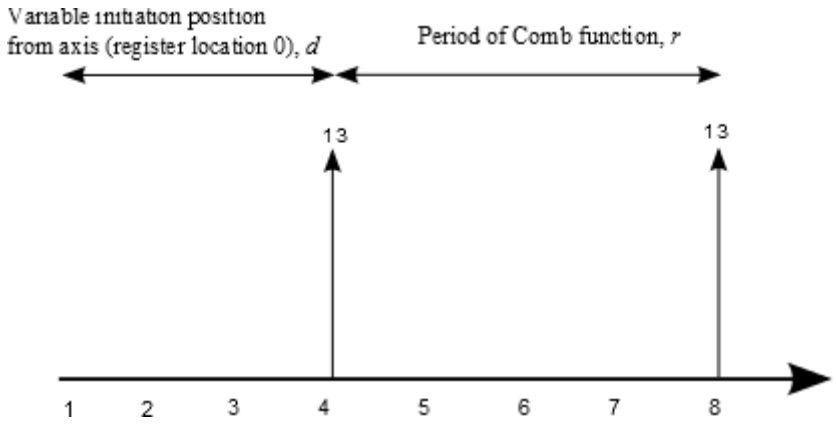

Figure 1(b). Suppose a measurement is made on the output register and the wavefunction collapses, for example, to the value 13. The entangled first register then partially collapses to a superposition state consistent with this measurement, as shown above.

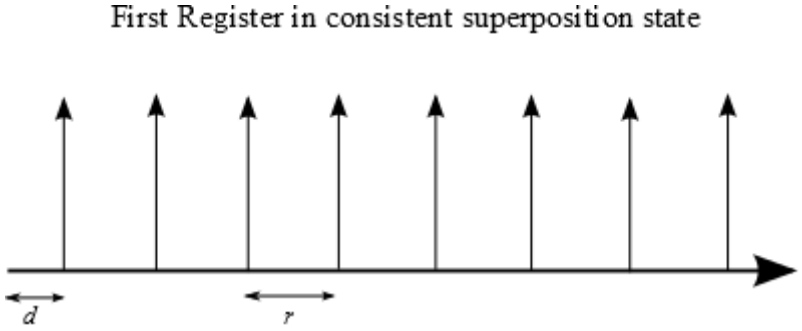

Figure 1(c). The first register thus consists of a *comb* function with period *r*, corresponding to the order, but with variable off-set *d* (dependant on which value the superposition state in the output register collapsed to: 1, 7, 4 or 13).

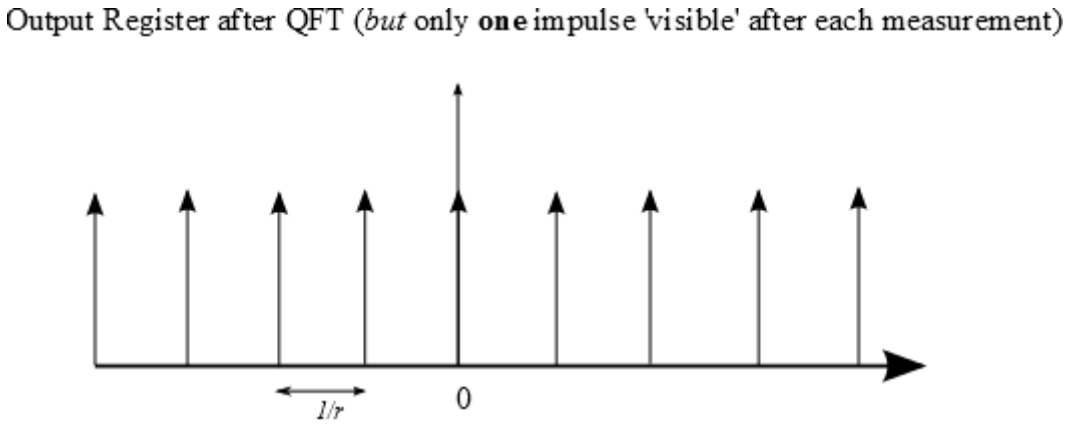

Figure 1(d). A QFT is performed on the first register state. This produces a centred reciprocal *comb* function with period 1/*r*. (A linear phase factor proportional to the initial *comb* function shift, *d*, is present but this does not affect the wavefunction measurement, $|\psi|^2$). Repeated runs of the algorithm result in a consistently spaced *comb* function in the first register and thus the reciprocal period (and hence the period, *r*) can be determined from repeated measurements on the first register after each run of the algorithm.

3. THE QUANTUM FOURIER AND HADAMARD TRANSFORMS

The critical role of the QFT in the Shor algorithm is thus apparent. A general problem in quantum computing is that calculations may be performed in a superposition state but to extract information from the calculation requires a measurement to be made and so a collapse of the wavefunction. The Shor algorithm ensures the solution to the problem has a periodicity which can be Fourier transformed to uniform peaks in the wavefunction which have a high probability of detection when a measurement is made. Thus useful information can be extracted from a superposition state.

This is possible because the Fourier transform operation is unitary i.e. the discrete Fourier transform matrix has the property that $\mathbf{W}\mathbf{W}^{\dagger} = \mathbf{I}$ (where the dagger superscript indicates the conjugate transpose of **W**). This implies that any physical realisation of the computational operation is reversible and non-dissipative[13]. In quantum computing, the unitary transformation implies conservation of the overall probability of detection of the propagated wavefunction through the system with only the distribution of the probability of detection over the output detector being altered from that of the input, the overall probability of collapse remaining conserved. Importantly, the wavefunction must propagate through the system without collapse, a requirement fundamental to quantum computing hardware.

Thus a QFT must be performed on the wavefunction generated by the modular arithmetic section of the algorithm. Let us assume we have, for example, four entangled qubits (i.e. a one *and* a zero superposition at each bit location) described by the wavefunction (or state vector):

$$|\psi\rangle = \sum_{n=0}^{N-1} x_n |n\rangle = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{4}$$

that is, the wavefunction can be considered a superposition of four qubits each weighted by a probability x_n representing the value of the input signal at that qubit location:

$$|\psi\rangle = x_{00}|00\rangle + x_{01}|01\rangle + x_{10}|10\rangle + x_{11}|11\rangle \tag{5}$$

The QFT of this state vector may then be computed:

$$|\Psi\rangle = \sum_{n=0}^{N-1} e^{-j\frac{2\pi mk}{N}} |\psi\rangle \tag{6}$$

which since this operation is unitary maintains the wavefunction superposition state but transforms it to the Fourier coefficients corresponding to input wavefunction, thus:

$$|\Psi\rangle = X_{00}|00\rangle + X_{01}|01\rangle + X_{10}|10\rangle + X_{11}|11\rangle \tag{7}$$

where the X_n are complex coefficients corresponding to the complex Fourier components at that qubit location in the output array. However, since they comprise the overall wavefunction they will not be directly accessible to measurement. Rather the probability of detection, by a single measurement, will be given by $|\Psi|^2$. If the input wavefunction is periodic, e.g. arises from the successful application of the Shor modular exponentiation procedure, the output wavefunction will have multiple peaks in its probability distribution at the output register locations corresponding to the periodicity of the output *comb* function, as described in Section 2. Thus repeated application of the QFT will yield more detection events at these locations and hence allow determination of the periodicity, $1/r$, of the *comb* function as required to factor the initial input sequence, N. Thus the QFT is more powerful than the FFT in that it can process 2^n inputs in parallel with effectively the same complexity of hardware structure (and so is exponentially faster in computation), as described below. However, the FFT yields N complex frequency components at its output whereas the QFT produces a probability distribution only which collapses to a single detection event upon measurement. This, however, may be very useful and used to solve problems with exponential complexity if applied appropriately, as has been shown in its application to the factoring problem.

The building block for the QFT is the Hadamard gate. This can be written:

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8}$$

This is the basic operation comprising the Hadamard transform and will act on a single qubit state to give:

$$\mathbf{H}\left[x_0|0\rangle + x_1|1\rangle\right] = \frac{1}{\sqrt{2}}(x_0 + x_1)|0\rangle + \frac{1}{\sqrt{2}}(x_0 - x_1)|1\rangle \tag{9}$$

Thus it can be seen that the Hadamard transform performs a 2-point QFT by implementing the basic FFT building block of the Butterfly operation i.e. the subtraction and addition of the two input signals, albeit in a superposition state[1].

Thus the FFT decomposition of the QFT can be made using the Hadamard gate as a basic building element together with controlled phase rotation gates described by matrices of the form[7,13]:

$$R_n = \begin{bmatrix} 1 & 0 & 0 & . & 0 \\ 0 & 1 & 0 & . & 0 \\ 0 & 0 & 1 & . & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & . & e^{-j\frac{2\pi}{2^n}} \end{bmatrix} \tag{10}$$

This allows the QFT structure for a *n*-qubit input to be represented compactly as shown in Figure 2.

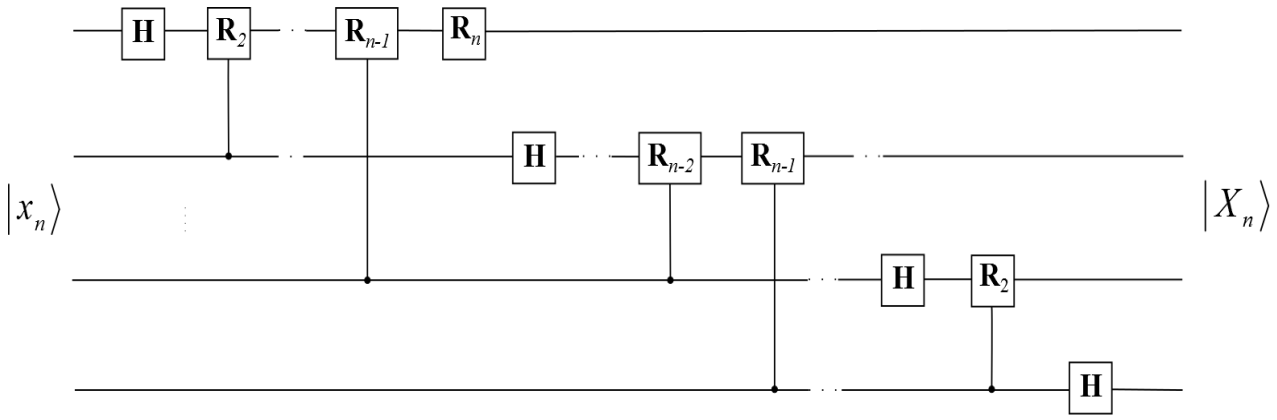

Figure 2. *n*-qubit QFT quantum gate array structure

The decomposition is thus an FFT structure acting on a binary *n*-qubit input to yield a complex-valued *n*-qubit output (in reverse bit order). However, an implementation problem arises in the phase resolution required by the final term in the R_n matrices for large *n*, as this scales as 2^n. Clearly, for a large *n* this would require impractically fine control of the phase. However, Coppersmith[14], by performing an analysis specifically for a radix-2 based FFT, has shown that for $n = 500$ less than a 0.5% reduction in the probability amplitude of the calculated spectral components can be obtained by limiting the quantization of the phase rotations to $\pi/64$. As can clearly be seen from equation (6) describing the QFT, this corresponds to reducing the number of discrete phase steps of the exponential function unit vector around the unit circle in the complex plane from 2^n to just 64 increments which in turn quantizes the cosine and sine basis functions each to 32 levels (due the projection of the unit vector in the complex plane onto the real and imaginary axes).

In an extensive simulation, Fowler and Hollenberg[15] have shown it may be possible to QFT a binary integer of several thousand bits with a phase control no better than $\pi/64$ and maintain good detection probabilities. However, they find that when the phase quantization is reduced further, a reduction of performance occurs.

The *N*-point Hadamard transform (sometimes known as the Walsh-Hadamard transform as it employs Walsh functions as its basis) may be written[10-12]:

$$H(k) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} h(m)(-1)^{b(k,m)} \qquad 0 \le k \le N-1 \tag{11}$$

where: $b(k,m) = \sum_{i=0}^{n-1} k_i m_i$; $k_i, m_i = 0,1$

and $\{k_i\}$ and $\{m_i\}$ are the bit states of the binary representations of k and m, respectively, to generate the binary amplitude Walsh basis functions of the HT[11].

The Hadamard transform (HT) is also a unitary transform and so can employed as part of a quantum algorithm. Indeed, the 2 x 2 Hadamard function plays a central role in generating qubit superposition states as described above. In general, the full HT N by N array can be decomposed into smaller sparse matrices as for the DFT and so a decomposition made similar to that for the fast Fourier transform (FFT) algorithm. The signal flow diagram for the fast HT is thus composed of butterfly operations similar to those of the FFT (for a decimation in time or decimation in frequency implementation) but without the requirement for the additional phase rotation factors necessary in the FFT since only 0 or π phase shifts are needed and these are inherent in the butterfly structures themselves[12]. The HT may thus be considered as the limit of the phase quantization process proposed by Coppersmith[14] and further analysed by Fowler and Hollenburg[15].

The HT performance when compared to that of the DFT depends strongly on the function to which it is applied. Clearly, it will not produce a single harmonic component if applied to a pure harmonic signal as would the DFT (assuming there are no spectral leakage effects effecting the DFT result i.e. a whole number of harmonic terms fit within the data window of length N employed). Of particular importance to the discussion here is that the HT will transform a *comb* function impulse train perfectly to a reciprocally spaced *comb* function[9-12]. Further similarities to the DFT are also realized, in particular, translations of the *comb* function will also generate a centred *comb* function in the reciprocal space but with 0 or π rotations of the appropriate individual impulse terms (rather than a more finely quantized linear phase term which the DFT produces). However, this, as required by the unitary nature of the HT, allows recovery of the original displaced *comb* function when the inverse HT is applied.

Thus the HT appears to have the desired properties for the *very* specific requirement of the Shor algorithm, i.e. the transformation of the *comb* function resulting from the partial collapse of the first quantum register in response to a measurement made on the output register which before measurement holds in a superposition state the results of the modular exponentiation operations, $x^a \bmod N$, $a = 0$ to q-1, as detailed in Section 2. There must, therefore, be an additional disrupting factor that prevents results much better than those predicated by the simulation work reported by Fowler and Hollenberg[15]. This factor appears to be due to a spectral leakage phenomenon affecting the accuracy of the HT transform. As has been explained in Section 2, the modular exponential function is periodic. However, and most importantly, this period will not, in the general case, divide the overall period of the q-1 samples of the data window. Since the modular exponential is effectively a sampled function (from $a = 0$ to q-1) it must be considered, as a well known consequence of the data sampling effect, to be periodically extended beyond the bounds of the data cell[9]. Thus, if a whole number of periods of the modular exponential function are not contained within this data window, a discontinuity will result at the data cell boundaries which, in turn, will lead to spectral leakage effects being apparent in the transformed data array, an effect that is well known in signal processing[9]. This will result in the reciprocal *comb* function (produced by the HT from the first register *comb* function) being disrupted. Thus rather than being a sequence of precisely located impulse functions there will be a spreading of spectral energy around each impulse which will in turn adversely affect the probability of the wavefunction collapsing to the precise locations of the data impulses comprising the reciprocal *comb* function which is required to determine the impulse train periodicity and hence factor the integer N.

A well-known signal processing technique to reduce spectral leakage is the application of a windowing function to the signal prior to transformation to the spectral domain[9-11]. Window functions smoothly attenuate the signal samples in the data window towards its boundaries, usually to zero, to force a periodicity of the data within the sampling window. In this way, the data samples at the cell extremities are matched to zero with the -1 and q^{th} samples i.e. the last and first samples of the preceding and following data cells. Thus the data samples are contained within an overall envelope function that is now periodic within the q-point data cell. Thus the window function would pre-multiply the *comb* function prior to transformation. A suitable windowing function might be the Bartlett window as this is a triangular function[10]:

$$w(m) = \begin{cases} \dfrac{2m}{q-1}, & 0 \le m < \dfrac{q-1}{2} \\ 2 - \dfrac{2m}{q-1}, & \dfrac{q-1}{2} \le m \le q-1 \end{cases}$$

(12)

that is a function that has a linear dependence on index q and so could be applied to the register contents with a Single Instruction Multiple Data (SIMD) instruction (in a similar manner to the modular exponentiation operations to produce x^a mod N). However, the operation must be unitary and so the *comb* function impulse train cannot be attenuated. Thus, rather than attenuating the impulse train by multiplying by $w(m)$, we instead add $w(m)$ to the comb function. This boosts the central terms in the data window, increasing their relative weight, and so will ameliorate the effect of a non-periodic fit of the data in the window. Since the *comb* function will not fall to zero at the window edges, this will not be as effective in reducing spectral leakage as multiplying by $w(m)$ but will still aid in reducing its deleterious effects. This, in turn, will increase the probability of detecting the reciprocal *comb* function impulse locations since they will be less spread and be more sharply peaked due to the reduction in spectral leakage effects. Thus the probability of the collapse of the output wavefunction, $|\psi_3>$, at one of the correct locations will be increased and a more reliable result achieved.

4. CONCLUSION

This paper has described the Shor quantum algorithm for large integer factorisation from a signal processing perspective, emphasising the importance of the quantum Fourier transform as an essential component of the algorithm. A problem will arise in the implementation of the QFT for large integer sizes as the phase control gates necessary for its exact implementation require, nominally, an exponentially fine phase control. Approximations to the QFT that would allow a much coarser phase control are reviewed. A quantum version of the Hadamard transform, requiring only a 0 or π phase control, is considered as this may provide an effective solution for the specific requirement of the Shor algorithm, namely the transformation of a *comb* function to a reciprocal *comb* function. The reasons for a possible degradation of performance of the Hadamard transform have been discussed and a possible means to ameliorate this degradation considered by applying a triangular windowing function prior the Hadamard transformation to effect a reduction in spectral leakage effects and so increasing the probability of accurate location of the output impulse function train upon measurement of the output wavefunction.

REFERENCES

1. Young R., Birch P., Chatwin C., "Coherent optical implementations of the fast Fourier transform and their comparison to the optical implementation of the quantum Fourier transform", Proc SPIE Vol 87480, 874806-1,-11, (2013).
2. Young R., Birch P., Chatwin C., "Considerations for extension of coherent optical processors in the quantum regime", Proc SPIE Vol 9845, 98450K, (2016).
3. Rivest, R., Shamir, A., Adelman, L., "A methods for obtaining digital signatures and public-key cryptosystems", Commun. ACM, Vol. 21, pp. 120-126, (1978).
4. Shor P. W., "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", SIAM J. Comput., Vol. 26(5), pp. 1484-1509, (1997).
5. Martín-López, E., Laing A., Lawson T., Alvarez R., Zhou X-Q., O'Brien J., "Experimental realization of Shor's quantum factoring algorithm using qubit recycling", Nature Photonics, Vol. 6, p. 773, (2012).
6. Lekitsch B., Weidt S., Fowler A., Molmer K., Devitt S., Wunderlich C., Hensinger W., "Blueprint for a microwave trapped ion quantum computer", Sci. Adv. Vol. 3, e1601549 pp.1-11, (2017).
7. Ekert A., Jozsa R., "Shor's quantum algorithm for factorizing numbers", Rev. Mod. Phys., Vol. 68, pp. 733-753, (1996).

8. Barak R., Ben-Aryeh Y., "Quantum fast Fourier transform and quantum computation by linear optics", J. Opt. Soc. Am. B, Vol. 24(2), pp. 231-240, (2007).
9. Brigham E., The Fast Fourier Transform, Prentice-Hall Inc, First Edition, (1974).
10. Pratt W., "Digital Image Processing", John Wiley & Sons, (1978).
11. Jain, K., "Fundamentals of digital image processing", Prentice-Hall Inc., (1989).
12. Beauchamp, K., "Applications of Walsh and related functions", Academic Press, (1984).
13. Nielsen M., Chuang I., "Quantum Computation and Quantum Information", 2nd Ed., Cambridge UP, (2001).
14. Coppersmith D., "An approximate Fourier transform useful in quantum factoring", IBM Research Report No. RC19642, T.J. Watson Research Center, Yorktown Heights, New York, (1994) (unpublished).
15. Fowler, A., Hollenberg L., "Scalability of Shor's algorithm with limited set of rotation gates", Physical Review A, Vol. 70, 032329-1:7, (2004).