Abstract Submitted
for the MAR13 Meeting of
The American Physical Society

Compilted Quantum Factoring Circuits OMAR GAMEL, DANIEL JAMES, University of Toronto — Shor's factoring algorithm is held as one of the most promising and useful applications of quantum computing. It allows one to factor large numbers in polynomial time, undermining the most common cryptographic schemes in use today, such as RSA cryptography. The well known algorithm is based on the quantum fourier transform to find the period of a function, and also makes heavy use of the modular exponentiation operation, given by,

$$U : a0 \rightarrow ax^a(modN), \tag{1}$$

where N is the number to be factored, and x is a random positive integer coprime with N. The modular exponentiation is the bottleneck of the algorithm, the portion that uses the most time. The generic algorithm can factorize any N in time order $(\log N)^3$, assuming sufficient memory space for intermediate calculations. Reducing the memory available (as long as it still lies above a certain threshold) increases the time taken by multiplicative factors, keeping its order the same in $\log(N)$. However, for a given N, or class of N's to factorize, the generic algorithm may be suboptimal, and can be optimized to result in substantial savings in both memory needed and operation time. The different suboperations involved in modular exponentiat

Omar Gamel
University of Toronto

Date submitted: 19 Nov 2012 Electronic form version 1.4