Porting algorithms from classical computers to quantum computers

Haiyue Kang[1], Charles Hill[1, 2]

1. Department of Physics, University of Melbourne, Australia

2. Department of Mathematics and Statistics, University of Melbourne, Australia

IBM Quantum Network Hub at the University of Melbourne

LABY FOUNDATION PTY LTD

ABSTRACT

This work presents a variety of implementations of quantum algorithms solving problems that can be easily done on classical computers, which included integer addition, multiplication, and exponentiation. In this work, we constructed an integer adder within four qubits and eight qubits respectively utilizing a known modification to the ripple-carry quantum addition algorithm. Moreover, it includes an integer multiplier of two arbitrary numbers within four qubits based on Quantum Fourier Transform and repeated addition as the essence of multiplication. Furthermore, established from our previous multiplier, this work presents an integer exponentiator that can perform exponentiation for arbitrary base and exponent limited to two and three qubits respectively. Each of these circuits is accomplished in a single unitary quantum circuit without classical measurement, making sure that number basis states or superposed states can be used as input without the generation of excess entropy.

1. INTRODUCTION

While it has been proven that some quantum algorithms can realize an exponential reduction in the order of resources (timesteps, number of qubits) compared to classical ones, most algorithms and techniques are implemented and optimized for classical computers, not for quantum computers. To address this issue, this work focuses on porting some basic arithmetic algorithms from classical computers to quantum computers in quantifying the resources required for arithmetic calculation on a quantum computer.

In the first step, by making use of the algorithm discovered by Gidney[1], which is based on a modification of ripple-carry adder[2], we build a quantum adder that is capable of performing integer addition within four and eight qubits respectively.

Then, following the implementation of the Quantum Fourier Transform[6] and its Hermitian conjugate as a basic building block, we managed to achieve multiplication of two arbitrary integers each within four qubits (as well as the case when one is fixed, here we choose number three) on a quantum circuit via repeated addition by controlled rotation about the *z*-axis on Bloch sphere between QFT and inverse QFT[3][4].

In the last part, the previous work enabled us to perform exponentiation of arbitrary base of two qubits (as well as the case of a fixed base, here we choose number three) and exponent of three qubits via repeated multiplication of the number in the base controlled by the qubits storing information about the exponent.

2. METHODOLOGY

- 2.1 Adder

To perform addition quantum mechanically, the design of the circuit includes two registers $|t\rangle = \sum_{k=0}^{n-1} |t_k\rangle$ and $|i\rangle = \sum_{k=0}^{n-1} |i_k\rangle$ each store a number being added in binary. There is an ancilla register $|a\rangle$ that acts as carry bits that calculate the product of two qubits $|x\rangle$ and $|y\rangle$, $|x\rangle \otimes |y\rangle$. That is, a logical-AND gate (shown in Fig 2) to decide whether incrementation should happen on the next digit. Since such operation would change the phase what *x* and *y* originally are and a carry bit contains information about the sum, to avoid wavefunction collapse, it is then 'un-computed' to restore qubits *x*, *y*, and carry bit. Next, the sum of two numbers can be easily determined by CNOT gate and stored as $|s\rangle$ (shown in Fig 1).

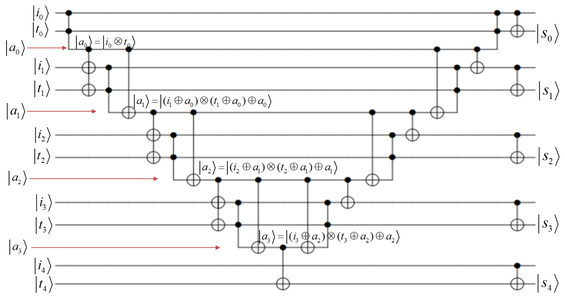

Fig 1: A 5-bit adder with 4 logical AND and 4 'un-computation'. Gidney, C. (2018). Adapted from: *Figure* 1, Halving the cost of quantum addition. Quantum, 2. https://doi.org/10.22331/q-2018-06-18-74)

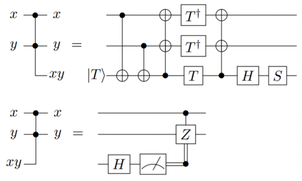

Fig 2. The design of logical AND operation and its erasure. Note although the second part undoes *x* and *y*, it is not unitary due to the measurement. This figure is taken from: Gidney, C. (2018). *Figure 3*, Halving the cost of quantum addition. Quantum, 2. https://doi.org/10.22331/q-2018-06-18-74)

- 2,2 Multiplier

In our circuit, the circuit is initialized with two registers $|a\rangle$ and $|b\rangle$ which stores the numbers *a* and *b* being multiplied in binary, a controlled qubit that decides whether an addition should be performed, as well as an accumulator initialized to zero state.

As mentioned previously, the key to a quantum multiplier is established by two ingredients: the Quantum Fourier Transform[6] and controlled repeated addition. A QFT will first apply to the accumulator, transforming them into phase space, followed by controlled addition[3, 4] acts repeatedly in the form of phase gate rotates about *z*-axis, and ends with inverse QFT(as shown in Fig 3). The logic of deciding what number should be added is simply directed by which two digit of *a* and *b* are multiplied together:

$$ab = \sum_{i=0}^{n-1} 2^i a_i \sum_{j=0}^{m-1} 2^j b_j = \sum_{i,j} 2^{i+j} a_i b_j$$

For example, if the qubit represents a_0 and b_1 are both 1, then this implies their product in decimal is $1\times 2^0 \times 1 \times 2^1 = 2$ which is the number should be added.

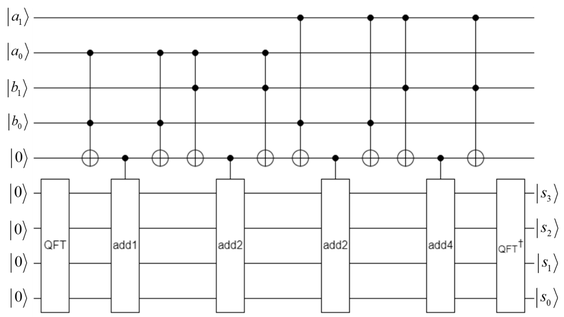

Fig 3: Illustration of a 2-bit by 2-bit multiplier that multiplies numbers *a*, *b*, and store as $|s\rangle$ by making use of the Quantum Fourier Transform and controlled addition via rotation on the z-axis. Note for a fixed number multiplier, say 3 times $|a\rangle$, we just need one register for $|a\rangle$ that controls the addition. The number in the repeated addition can be decided by 'memorising' the value of $3a_0$, $3a_1$ etc. and encoded applied gates.

- 2.3 Exponentiator

Just like multiplication is repeated addition, exponentiation is just repeated multiplication. We make our exponentiator by nesting the multiplier we made into the circuit. To calculate a^x, the circuit consists of two registers storing $|a\rangle$ and $|x\rangle$, as well as an accumulator initialized to $|1\rangle$. Multiplication of *a*, a^2, a^4 etc. is acted on the accumulator controlled by a qubit representing each digit of *x*. Note *a*, a^2, a^4, etc. are also calculated by repeatedly squaring by itself (as shown in Fig 4). The framework was inspired by the quantum multiplier constructed in Rines, R., & Chuang's paper[5]. Also, a set of unitary inverse operations that undo the working qubits no longer used are appended to the circuit to minimize error caused by collapse of states during measurements.

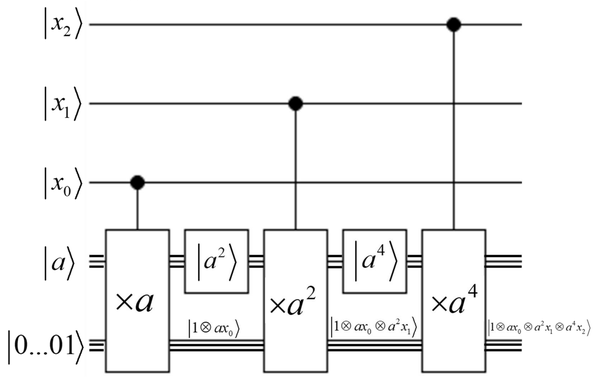

Fig 4: Exponentiator with 3-qubit $|x\rangle$. The gates for multiplication have a structure the same as the multiplier defined previously. For the squaring gate, it multiplies by itself and stores the output back on the same register. For the controlled multiplication gate, it stores the output to the accumulator. Note the register for *a* and accumulator is represented as a bundle of qubits and the actual number of qubits depends on requirements. If one wants to construct an exponentiator for a fixed number, say 3, simply remove the register of *a* and utilise the pre-computed value of the numbers 3, 3^2, 3^4 in controlled multiplications.

3. RESULTS

The operations of this project are successfully demonstrated as codes. They can be found on GitHub, please refer to: https://github.com/KangHaiYue/Laby-Research-Project-Quantum-computing

Fortunately, our circuits with 42 as the maximum number of qubits should be able to be performed on the latest quantum computer (127 qubits[7]). However, considering the total number of two or more qubits gates in our circuits is at least 22 (as shown in Table 1), the timesteps required for all circuits will be much more than 10. Therefore, the QV required for all circuits will be exceeds the maximum QV current quantum computers offer today (1024QV[8]). This implies that there will be significant errors when the circuits are running on the actual machine.

Circuit name	Qubits	Toffolis	Phase gates $\leq \frac{\pi}{4}$	CNOTs
Adder_nibble (4 qubits)	12	0	12	33
Adder_byte (8 qubits)	23	0	28	81
Multiplier of 3	10	0	22 (controlled)	0
General multiplier	20	32	90 (controlled)	0
Exponentiator of 2	23	24	72 (controlled)	12
General exponentiator	42	249	469 (controlled)	26

Table 1: Qubit and gate counts for each circuit

4. CONCLUSION

We used Gidney's algorithm to demonstrate quantum circuits that perform the addition of four and eight qubits. We also designed a quantum multiplication circuit for arbitrary two numbers within four qubits and fixed number respectively, as well as the circuit that can raise *a* to the power of *x* where *a* can be either fixed or arbitrary within two qubits, *x* within three qubits. Although the most advanced quantum computers have enough qubits to perform those circuits, significant errors would be generated.

5. ACKNOWLEDGEMENTS

I would like to thank the Laby Foundation which supported this work under the Laby Research Scholarship Program and am grateful for the help from my supervisor Dr. Charles Hill. I would also like to thank the IBM Quantum Hub at the University of Melbourne for technical support.

REFERENCES

[1] Gidney, C. (2018). Halving the cost of quantum addition. *Quantum*, *2*. https://doi.org/10.22331/q-2018-06-18-74

[2] Cuccaro, S. A., Draper, T. G., Kutin, S. A., & Moulton, D. P. (2004). *A new quantum ripple-carry addition circuit*, https://arxiv.org/abs/quant-ph/0410184

[3] White, G. A. L., Hill, C. D., & Hollenberg, L. C. L. (2021). *Truncated phase-based quantum arithmetic: error propagation and resource reduction*. http://arxiv.org/abs/2110.00217

[4] Ruiz-Perez, L., & Garcia-Escartin, J. C. (2017). Quantum arithmetic with the quantum Fourier transform. *Quantum Information Processing*, *16*(6). https://doi.org/10.1007/s11128-017-1603-1

[5] Rines, R., & Chuang, I. (2018). *High Performance Quantum Modular Multipliers*.

[6] Nielson, M., & Chuang, I., (2010, 10th edition) Quantum Computation and Quantum Information

[7] IBM Newsroom. 2021. *IBM Unveils Breakthrough 127-Qubit Quantum Processor*. [online] Available at: https://newsroom.ibm.com/2021-11-16-IBM-Unveils-Breakthrough-127-Qubit-Quantum-Processor

[8] Honeywell. 2021. *Honeywell Sets Another Record For Quantum Computing Performance*. [online] Available at: https://www.honeywell.com/us/en/news/2021/07/honeywell-sets-another-record-for-quantum-computing-performance