

A QUANTUM ALGORITHM TO FIND THE MAXIMUM OF A PAIR OF (SIGNED) INTEGERS

The idea for the code came from a video by Anant Vigyan, see [1].

Comments are welcome (email to akosnagymath@gmail.com).

1. THE $U_{<}$ GATE

First, let's construct a quantum gate, $U_{<}$, on 3 quantum qubits with the following property: given two classical bits, $a, b \in \{0, 1\}$, the effect of $U_{<}$ on $|ab0\rangle (= |a\rangle|b\rangle|0\rangle)$ is

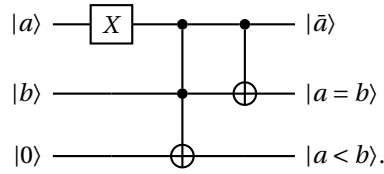
$$U_{<}(|ab0\rangle) = |\bar{a}(a = b)(a < b)\rangle,$$

where $\bar{a} := \text{NOT } a$.

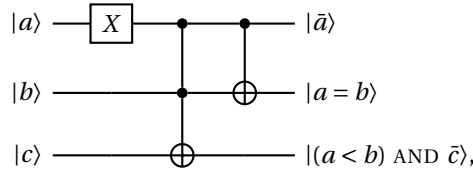
For the rest of this document, $\text{CNOT}_{i,j}$ will denote the CNOT gate with control qubit i and controlled qubit j . Similarly $\text{TOFF}_{i,j,k}$ will denote the Toffoli/TOFF gate with control qubits i and j , and controlled qubit k . Abstractly, the gate can be given as

$$U_{<} = (\text{CNOT}_{1,2} \otimes \mathbb{1}_2)(\text{TOFF}_{1,2,3})(X \otimes \mathbb{1}_2 \otimes \mathbb{1}_2),$$

where $\mathbb{1}_2$ is the 2-by-2 identity matrix. Schematically $U_{<}$ is



More generally, the effect of $U_{<}$ for any $a, b, c \in \{0, 1\}$ bits is



or, more abstractly, $U_{<}(|abc\rangle) = |\bar{a}(b + \bar{a})(c + \bar{a}b)\rangle$, where addition and multiplication is understood modulo 2, and thus, in the lexicographically ordered computational basis¹, the matrix of $U_{<}$ is

$$(U_{<}) = \begin{pmatrix} 0_4 & \mathbb{1}_4 \\ X \otimes \mathbb{1}_2 & 0_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

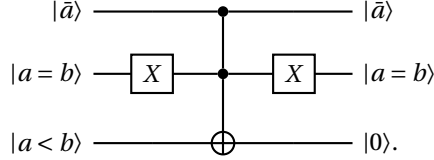
The claim about the effect of $U_{<}$ is now a matter of simple computation.

¹that is, in $\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}$

Remark 1.1. Note that $U_{<}$ returns two important pieces of information: 1. which bit is larger (or equal to) than the other one, and 2. whether the bits are equal. The algorithm makes use of this by first implementing $U_{<}$ (with $c = 0$) to each digit (including the sign digit) of number_1 and number_2 .

2. THE U_0 GATE

The purpose of this gate is to reset the third qubit to $|0\rangle$ in the output of $U_{<}$. This can easily be implemented with



3. THE CLASSICAL ALGORITHM

Next, I describe the idea behind the algorithm.

We represent a nonnegative integer, $\text{number} \in \mathbb{N}$ via

$$\text{number} = {}_n 1c_1 \dots c_n,$$

where $n \geq \max(1, \lceil \log_2(\text{number}) \rceil)$, $c_i \in \{0, 1\}$, and $(c_1 \dots c_n)_2 = \text{number}$. For negative integers, $\text{number} \in -\mathbb{N}_+$, use

$$\text{number} = 0c_1 \dots c_n,$$

where $n \geq \max(1, \lceil \log_2(|\text{number}|) \rceil)$, $c_i \in \{0, 1\}$, and $(c_1 \dots c_n)_2 = 2^n - |\text{number}|$. This is **not** the usual binary representation of signed integers, but it is useful for the problem at hand. In particular the Python code:

```
bits = []                                # array to store the bits

sign = number >> (32 - 1)                # 0 if number is positive, - 1 if negative
                                           # note: this works Python's (signed) 'int' type
                                           # for other types the power 32 might need to be replaced

while number != sign:
    bits.append([number & 1])             # appends bits with the last binary digit
    number >>= 1                          # discards the last digit

bits = bits[::-1]                        # reverses bits so that it starts with c_0
```

produces the array $\text{bits} = [c_0, c_1, \dots, c_n]$, with $n = \lceil \log_2(|\text{number}|) \rceil$. This has $O(\log_2(|\text{number}|))$ space and time complexity. If number is given in other binary representations, the conversion can also be done with the same space and time complexities.

Let now $n := \max(1, \log_2(\max(|\text{number}_1|, |\text{number}_2|)))$ and assume that the inputs, number_1 and number_2 , are given the forms

$$\text{number}_1 = a_0 a_1 \dots a_n,$$

$$\text{number}_2 = b_0 b_1 \dots b_n,$$

defined above.

Since nonnegative numbers are larger than negatives, the statement $((\text{number}_1 < \text{number}_2) \text{ AND } (b_0 < a_0))$ is False. Thus we have the following identity, by the virtue of our binary presentation

$$\begin{aligned} (\text{number}_1 < \text{number}_2) &= (a_0 < b_0) \text{ OR } (a_1 \dots a_n < b_1 \dots b_n) \\ &= (a_0 < b_0) \\ &\text{OR} \\ &((a_1 < b_1) \text{ XOR } (a_1 = b_1) \text{ AND } (a_2 \dots a_n < b_2 \dots b_n) \text{ XOR } \dots) \end{aligned}$$

In the next, final section, I construct a quantum circuit and show that a particular qubit is always found in the eigenstate $|\text{number}_1 < \text{number}_2\rangle$.

4. THE QUANTUM CIRCUIT

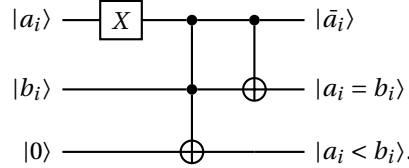
Again assume that the inputs, $\text{number}_1, \text{number}_2 \in \mathbb{Z}$, are given the forms

$$\text{number}_1 = a_0 a_1 \dots a_n,$$

$$\text{number}_2 = b_0 b_1 \dots b_n,$$

where $n := \max(1, \lceil \log_2(\max(|\text{number}_1|, |\text{number}_2|)) \rceil)$. The algorithm can be described as follows:

Step 0: Prepare $3 * n$ quantum registers and a classical register. For each $i \in \{0, 1, \dots, n\}$, initialize the $(3 * i + 1)$ st qubit as $|a_i\rangle$, the $(3 * i + 2)$ nd qubit as $|b_i\rangle$, and the $(3 * i + 3)$ rd register as $|0\rangle$. Attach a $U_{<}$ gate to each such triplet:



Since $n \geq 1$, This generates at least 6 registers.

Step 1: Attach a $\text{TOFF}_{2,6,3}$ gate, a U_0 gate to the registers 4, 5, and 6, and finally a $\text{TOFF}_{2,5,6}$ gate.

Note that right after this gate, the third register is in the state $|(a_0 < b_0) \text{ OR } (a_1 < b_1)\rangle$.

Step ≥ 2 : For $i \in \{2, \dots, n\}$ (in the normal order), attach a $\text{TOFF}_{3*i, 3*i+3, 3}$ gate, then a U_0 gate to the registers $(3 * i + 1)$, $(3 * i + 2)$, and $(3 * i + 3)$.

Note that right after this gate, the third register is in the state $|(a_0 < b_0) \text{ OR } (a_1 \dots a_i < b_1 \dots b_i)\rangle$.

Last step: Measure the third qubit, which is not in the state $|(a_0 < b_0) \text{ OR } (a_1 \dots a_n < b_1 \dots b_n)\rangle = |\text{number}_1 < \text{number}_2\rangle$.

Remark 4.1. The attachment of the last U_0 gate does not change the result of the measurement.

REFERENCES

- [1] Anant Vigyan, *Quantum algorithm 2 quantum comparison (new, to be published)*, available at <https://www.youtube.com/watch?v=AmqvNmzeRkA>. [†]1