Cuccaro Ripple-Carry Adder

Cody M. Grant

(Dated: February 11, 2022)

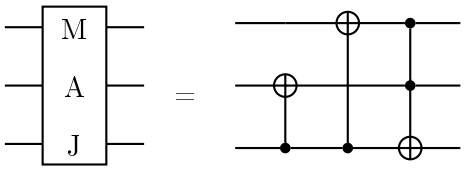
Abstract

The work for a ripple-carry addition circuit was done by Cuccaro, Draper, Kutin, and Moulton (arXiv: 0410184 [quant-ph]).

It implements 2 gates constructed out of CX, CCX, and X gates.

I. MAJ GATE

The first gate is the MAJ gate, which computes the majority of the three bits in place.



mathematically, the first 2 CX gates equate to

$$\begin{aligned}(CX(2,1))(CX(2,0)) &= (|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes X \otimes I) \\ &\quad \times (|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes I \otimes X) \\ &= (|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes X \otimes X) \end{aligned} \tag{1}$$

Then multiplying in the CCX gate

$$\begin{aligned}[\text{MAJ}] &= (|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes X \otimes X) \\ &\quad \times (I \otimes I \otimes |0\rangle\langle 0| + I \otimes |0\rangle\langle 0| \otimes |1\rangle\langle 1| + X \otimes |1\rangle\langle 1| \otimes |1\rangle\langle 1|) \\ &= |0\rangle\langle 0| \otimes [I \otimes |0\rangle\langle 0| + |0\rangle\langle 0| \otimes |1\rangle\langle 1|] \\ &\quad + |0\rangle\langle 1| \otimes [|1\rangle\langle 1| \otimes |1\rangle\langle 1|] \\ &\quad + |1\rangle\langle 0| \otimes [|0\rangle\langle 1| \otimes |0\rangle\langle 1|] \\ &\quad + |1\rangle\langle 1| \otimes [X \otimes |1\rangle\langle 0| + |1\rangle\langle 0| \otimes |0\rangle\langle 1|] \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{2}$$

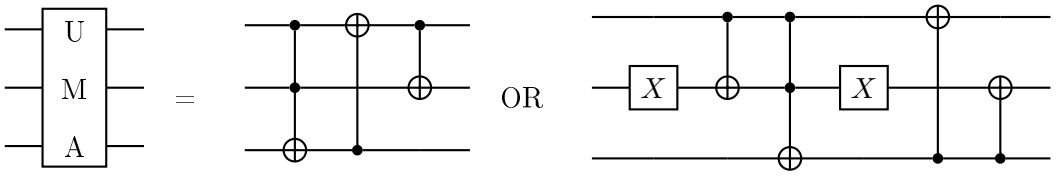
What we want to show is that after this gate: the first qubit $c_i \to c_i \oplus a_i$, the second qubit $b_i \to b_i \oplus a_i$, and the third qubit $a_i \to c_{i+1}$. And our above gate transforms the 3-state qubit wavefunction like:

$$|a_i b_i c_i\rangle = \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \xrightarrow[\text{[MAJ]}]{} \begin{pmatrix} 000 \\ 001 \\ 010 \\ 111 \\ 011 \\ 110 \\ 101 \\ 100 \end{pmatrix} = |c_{i+1}(b_i \oplus a_i)(c_i \oplus a_i)\rangle \tag{3}$$

and is this what we actually have? Yes, if you do it case-by-case, you find that this is true, although a little hard to wrap your head around initially if you aren't use to mod-2 addition. I am curious as to how they thought about constructing it. Perhaps, it was in tandem with the other gate they use, the UMA gate, but the whole process is really interesting to me.

II. UMA GATE

The second gate in their adder, which stands for the "UnMajority and Add" gate. They have two versions of this gate: the first being simpler to read, and the second has greater parallelism.



(d) 2-CX version

(e) 3-CX version

In both cases, they take the three incoming qubits and convert them the same way: $c_i \oplus a_i \to c_i$, $b_i \oplus a_i \to s_i$, and $c_{i+1} \to a_i$, where $s_i = c_i \oplus b_i \oplus a_i$. Since the result is the same, we can break down the simpler gate for ... uhh... simplicity. Starting with the two

CX gates:

$$\begin{aligned}(CX(2,0))(CX(0,1)) &= (|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes I \otimes X) \\ &\quad \times (I \otimes I \otimes |0\rangle\langle 0| + I \otimes X \otimes |1\rangle\langle 1|) \\ &= (|0\rangle\langle 0| \otimes (I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|) + |1\rangle\langle 1| \otimes (I \otimes |1\rangle\langle 0| + \otimes X \otimes |0\rangle\langle 1|))\end{aligned} \tag{4}$$

Multiplying with the Toffoli to make the UMA_2 gate:

Then multiplying in the CCX gate

$$\begin{aligned}[\text{UMA}] &= (I \otimes I \otimes |0\rangle\langle 0| + I \otimes |0\rangle\langle 0| \otimes |1\rangle\langle 1| + X \otimes |1\rangle\langle 1| \otimes |1\rangle\langle 1|) \\ &\quad \times (|0\rangle\langle 0| \otimes (I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|) + |1\rangle\langle 1| \otimes (I \otimes |1\rangle\langle 0| + X \otimes |0\rangle\langle 1|)) \\ &= |0\rangle\langle 0| \otimes [I \otimes |0\rangle\langle 0| + |0\rangle\langle 1| \otimes |1\rangle\langle 1|] \\ &\quad + |0\rangle\langle 1| \otimes [|1\rangle\langle 1| \otimes |1\rangle\langle 0|] \\ &\quad + |1\rangle\langle 0| \otimes [|1\rangle\langle 0| \otimes |1\rangle\langle 1|] \\ &\quad + |1\rangle\langle 1| \otimes [X \otimes |0\rangle\langle 1| + |0\rangle\langle 0| \otimes |1\rangle\langle 0|] \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\end{aligned} \tag{5}$$

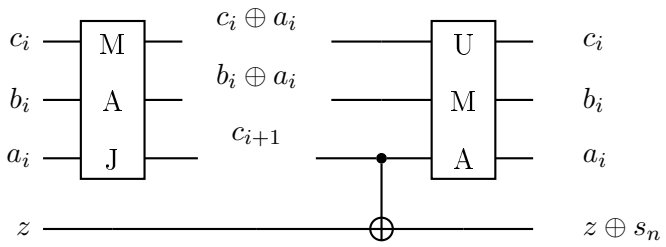
The initial wavefunc transition is

$$|c_{i+1}(b_i \oplus a_i)(c_i \oplus a_i)\rangle = \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \xrightarrow[\text{[UMA]}]{} \begin{pmatrix} 000 \\ 011 \\ 010 \\ 110 \\ 111 \\ 100 \\ 101 \\ 001 \end{pmatrix} = |a_i s_i c_i\rangle \tag{6}$$

I find it interesting that it almost entirely reverses the output of the MAJ gate, but the second qubit is no longer one of the digits we need to add, it is the i-th digit of the sum we are looking for.

III. ALL TOGETHER

We are looking to add two n-digit numbers together ($a + b = s$). We will find each individual digit of the sum, s_i, with each individual digit of the two numbers, $a_i + b_i$, plus the carry over bit for that digit, c_i, and therefore $s_i = a_i \oplus b_i \oplus c_i$ for $i < n$. The initial carry over bit is always $c_0 = 0$, and the final carry over bit is the final bit of the calculation $s_n = c_n$. Diagramatically, for $n = 1$, we find:



And mathematically, this would be multiplying the two gates together, which is a lot of terms in bra-ket notation. Now I visually like working with this notation, but with so many terms, to save me from typing so much, and for readability, I will do it with matrix multiplication

$$[\text{MAJ}][\text{UMA}] = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

The transformation of the wavefunction is then

$$|a_i b_i c_i\rangle = \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \xrightarrow[\text{[UMA]}]{} \begin{pmatrix} 000 \\ 011 \\ 010 \\ 110 \\ 110 \\ 101 \\ 100 \\ 111 \end{pmatrix} = |a_i s_i c_i\rangle \tag{8}$$

This circuit has many gates, and once the number of digits get larger, it has many, many gates. One thing that can be done is to analyze the circuit and see any redundancies and take those gates out or move many gates to one time slice, if applicable. This can be read about more in the paper referenced in the abstract. A main point being, this is the case where you

want to use the more complex UMA_3 gate as it allows for more of these simplifications to happen.
