Classical simulation of the QuEra circuit

Here we sketch a classical algorithm for computing output amplitudes of the QuEra circuit [1]. This circuit acts on n qubits where n is divisible by three. Let $m = n/3$. Define m-qubit registers R, B, G. These registers represent three species of logical qubits colored in red, blue, and green on Extended Data Fig. 6 of [1], see page 29. The main part of the QuEra circuit alternates between two types of gate layers. First, there are layers of CCZ and CZ gates. Qubits that participate in a CCZ or CZ gate must carry distinct colors. For example, each CCZ gate acts on a (red,blue,green) triple of qubits. A CZ gate may act on pairs (red,blue), (red,green), or (blue, green). Second, there are CNOT layers. The control and the target of each CNOT carry the same color. Thus only (red,red), (blue,blue), or (green, green) CNOTs are allowed. The circuit begins and ends with a Hadamard layer $H^{\otimes n}$. The above restrictions on CCZ/CZ/CNOT gates are dictated by the chosen error correcting code. Namely, this code only enables a fault-tolerant (transversal) implementation of CCZ/CZ/CNOT gates satisfying the above restrictions. The QuEra circuit has some additional structure (the hypercube topology) which is not dictated by quantum fault-tolerance and can be easily changed. Accordingly, here we ignore this additional structure.

We shall write n-bit strings as $x = (x^R, x^B, x^G)$ for m-bit strings x^R, x^B, x^G obtained by restricting x onto the red, blue, and green registers. Let R_i, B_i, G_i be the i-th qubit of registers R, B, G respectively.

Applying the first Hadamard layer to the initial state $|0^n\rangle$ is equivalent to initializing each qubit in the X basis. Thus our initial state is

$$|\psi_0\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

Let $|\psi_t\rangle$ be the state generated by the first t gates in the circuit (ignoring the initial and the final Hadamard layers). We claim that

$$|\psi_t\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{f_t(x)} |x\rangle$$

for some cubic phase polynomial $f_t : \mathbb{F}_2^n \to \mathbb{F}_2$ which is linear in each variable $x^R, x^B, x^G \in \{0,1\}^m$. To explicitly describe this polynomial, define a set of monomials

$$\mathcal{M} = \{x_i^R x_j^B x_k^G, \quad x_i^R x_j^B, \quad x_j^B x_k^G, \quad x_i^R x_k^G \,:\, 1 \le i, j, k \le m\}.$$

Then we claim that for each time step t there will be a subset of monomials $M_t \subseteq \mathcal{M}$ such that

$$f_t(x) = \sum_{h \in M_t} h(x) \pmod{2}. \tag{1}$$

Indeed, initially $t = 0$ and the phase polynomial is zero, $f_0(x) = 0$ for all x. Accordingly, $M_0 = \emptyset$. Applying a CCZ gate to a triple of qubits R_i, B_j, G_k modifies the phase polynomial as $f_{t+1}(x) = f_t(x) + x_i^R x_j^B x_k^G$. Here and below the addition of Boolean polynomials is modulo two. Thus

$$M_{t+1} = M_t \oplus \{x_i^R x_j^B x_k^G\}.$$

where $\oplus$ stands for the exclusive OR (symmetric difference) of two sets of monomials. Applying a CZ gate to pair of qubits R_i and B_j modifies the phase polynomial as $f_{t+1}(x) = f_t(x) + x_i^R x_j^B$. Thus

$$M_{t+1} = M_t \oplus \{x_i^R x_j^B\}$$

with a similar update rule for other combinations of colors. Applying a (red,red) CNOT with a control R_c and target R_τ is equivalent to a linear change of variables $x_\tau^R \leftarrow x_\tau^R + x_c^R$ (modulo two). The corresponding update rule for the phase polynomial can be described as

---

$M_{t+1} \leftarrow M_t$
for $h \in M_t$ **do**
    if h contains x_τ^R **then**
        Write $h = x_\tau^R h'$ for some monomial h'
        $M_{t+1} \leftarrow M_{t+1} \oplus \{x_c^R h'\}$
    end if
end for

A similar update rule applies to (blue,blue) and (green,green) CNOTs. The above takes care of all gates except for the final Hadamard layer. Let L be the number of gates in the QuEra circuit (not counting Hadamards) and $|\psi_L\rangle$ be the state reached immediately before the final layer of Hadamards. Let $f_L(x)$ be the corresponding phase polynomial. Our goal is to compute an amplitude $\Psi_{out}(s) = \langle s|H^{\otimes n}|\psi_L\rangle$ for some given bit string $s \in \{0,1\}^n$. In other words,

$$\Psi_{out}(s) = \langle s|U_{QuEra}|0^n\rangle,$$

where U_{QuEra} is the entire QuEra circuit. A simple algebra gives

$$\Psi_{out}(s) = \frac{1}{2^m} \sum_{x^R \in \{0,1\}^m} (-1)^{s^R \cdot x^R} \langle s^{BG}|H^{\otimes 2m}|BG(x^R)\rangle,$$

where $s^{BG} = (s^B, s^G)$ is the restriction of s onto blue and green registers and

$$|BG(x^R)\rangle = \frac{1}{2^m} \sum_{x^B, x^G \in \{0,1\}^m} (-1)^{f_L(x^R, x^B, x^G)} |x^B, x^G\rangle$$

is a state of the registers BG parameterized by x^R. The key observation is that the phase polynomial f_L with a fixed variable x^R becomes a quadratic function of x^B and x^G (because each monomial in f_L includes at most one variable of each color). Accordingly, $|BG(x^R)\rangle$ is a stabilizer state of $2m$ qubits. Such state can be created using only Clifford gates (Hadamards, CZ and Z). We can compute amplitudes $\langle s^{BG}|H^{\otimes 2m}|BG(x^R)\rangle$ efficiently using off-the-shelf Clifford simulators. Computing $\Psi_{out}(s)$ then amounts to carrying out 2^m Clifford simulations on $2m$ qubits. In fact, the Clifford circuit that generates $|BG(x^R)\rangle$ has a special structure that simplifies the Clifford simulation. Namely, consider some fixed $x^R \in \{0,1\}^m$. We can write

$$f_L(x^R, x^B, x^G) = \sum_{j,k=1}^m \Gamma_{j,k} x_j^B x_k^G + \sum_{j=1}^m \delta_j^B x_j^B + \sum_{k=1}^m \delta_k^G x_k^G \pmod 2$$

for some coefficients $\Gamma_{j,k}, \delta_j^B, \delta_k^G \in \{0,1\}$ that depend on x^R (we do not explicitly show this dependence to ease the notations). Below we consider Γ as a matrix of size $m \times m$. We consider δ^B and δ^G as column vectors of size m. Then

$$\langle s^{BG}|H^{\otimes 2m}|BG(x^R)\rangle = \frac{1}{2^{2m}} \sum_{x^B, x^G \in \{0,1\}^m} (-1)^{s^B \cdot x^B + s^G \cdot x^G + x^B \cdot \Gamma x^G + \delta^B \cdot x^B + \delta^G \cdot x^G}.$$

Here dot denotes inner product of binary vectors and Γx^G denotes matrix-vector multiplication. The above sum can be computed analytically. After simple algebra one gets

$$\langle s^{BG}|H^{\otimes 2m}|BG(x^R)\rangle = \begin{cases} 2^{-\mathrm{rank}(\Gamma)}(-1)^{(\delta^G+s^G)\cdot\Gamma^{-1}(\delta^B+s^B)} & \text{if} \quad \delta^B + s^B \in \mathrm{Col}(\Gamma) \quad \text{and} \quad \delta^G + s^G \in \mathrm{Row}(\Gamma) \\ 0 & \text{else} \end{cases} \tag{2}$$

Here we write $\mathrm{Col}(\Gamma)$ and $\mathrm{Row}(\Gamma)$ for the linear subspace of $\{0,1\}^m$ spanned by columns and by rows of Γ respectively. Finally, by a slight abuse of notations, we write $\Gamma^{-1}(\delta^B + s^B)$ for any solution $x^G \in \{0,1\}^m$ of the linear system

$$\Gamma x^G = \delta^B + s^B \tag{3}$$

(we do not assume that Γ is invertible). This system is feasible whenever $\delta^B + s^B \in \mathrm{Col}(\Gamma)$. The inner product $(\delta^G + s^G) \cdot \Gamma^{-1}(\delta^B + s^B) = (\delta^G + s^G) \cdot x^G$ in Eq. (2) is the same for all solutions x^G of the linear system Eq. (3) due to the condition $\delta^G + s^G \in \mathrm{Row}(\Gamma)$. Indeed, different solutions x^G of the linear system Eq. (3) differ by a vector from the nullspace of Γ. Such vector is orthogonal to any row of Γ.

To conclude, computing amplitudes Eq. (2) requires only the standard linear algebra over the binary field: computing the rank of a matrix and solving linear systems. The time complexity scales as $O(m^3)$. The overall computation of $\Psi_{out}(s)$ takes time $O(m^3 2^m)$.

References

[1] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. Logical quantum processor based on reconfigurable atom arrays. *arXiv:2312.03982*, 2023.