

To continue our discussion from the previous section, we now go into more details of how the idea of *modules* effectively captures a realistic resource in quantum computing. Using a strictly CCNTC model, either in 1D or 2D, makes architectures very sensitive to geometry. That is, the exact placement and distribution of data throughout a lattice can vastly change the resources needed to communicate and move data throughout such an arrangement. While such details are necessary for actual implementations, in preliminary studies we found that such communication formed the bulk of such communication for factoring on a 2D CCNTC architecture.

Our goal is to study more general architectural features that are not so sensitive to placement, or moreover, that obviate the need by treating each module as a separate quantum computer. The key insight of parallelization results such as Browne-Kasheffi-Perdrix and Kasheffi-Broadbent is that the power of communication between fixed-size, parallel quantum computers can greatly increase their overall computing power.

However, given the difficulty of establishing remote entanglement, we are limiting the communication between modules to be of constant size, although the modules themselves do not have to be constant-sized. For example, the rate of generating entangled photon pairs in current laboratory experiments is one every few seconds (citation needed, maybe from NIST), which is much larger than the coherence times of many physical qubits, including trapped ions. Therefore, generating scalable entanglement resources for communication between separate quantum computers seems a difficult task for the foreseeable future. This is the rationale behind the choice of restrictions behind the $O(1)$ communication between modules and the $O(n)$ size of the modules themselves, which otherwise might seem arbitrary.

Another subtlety of the model we wish to discuss is the concept of *reusable ancillae*, that is, a qubit which starts out initialized to $|0\rangle$ at the beginning of a circuit and is returned to $|0\rangle$ by the end of the circuit, where we use the circuit here loosely to mean the smallest sequence of operations in which an ancilla is used (entangled with the computation state) and then becomes disentangled again. Clearly, such an ancilla can be used multiple times, becoming entangled and then disentangled from the computation state through the lifetime of a complicated algorithm, each time performing a different purpose.

Unlike much of the previous literature, we do not treat such ancillae as having a fixed location and needing to be maintained in a coherent state throughout the life of the computation. Rather, it only has to be maintained for the above *minimal computation time*. Such an ancilla may be measured in the middle of a computation in order to make it a product state, or to disentangle it from the computation state, to free up qubits which would otherwise have to be maintained by a quantum error-correcting code. These ancillae, while they are in a $|0\rangle$ state, may even be dumped out of the quantum computer, for example, like ions that can be lost out of the trap, or moved to a different area of the trap where they are needed. The end result is that we treat such ancillae as instantly (re)-initializable and available in arbitrary locations within a module “just-in-time”. This is consistent with how we currently understand experimental

quantum information systems to be built. That is, in segmented ion traps, not all fabricated locations hold ions, only those that currently need to be involved in the computation. In superconducting systems, such as the D-Wave architecture, not all couplings need to be active. For those ancillae which lie dormant, all couplings to them from adjacent qubits can be turned off.

The idea of the computation state is one that persists over time, and has the following definition. In the last timestep of a circuit, the computation state is defined by the final qubits which are projectively measured to obtain a classical answer. At every timestep before that, say step i , the computation state is one which is connected to the computation state in step $i + 1$ by an entangling gate (most two-qubit gates, cite Nielsen et al paper). In analogy to classical computing, any qubit that is not part of the computation state at any timestep can safely be removed from the circuit without affecting the computation. It is similar to dead code analysis in a classical compiler, where code that cannot possibly affect the outcome of a function is pruned away.

The computation state is used to measure the resource of *circuit coherence*, the total amount of error-correction, or state maintenance, needed to complete a circuit. If at any timestep of a quantum circuit, one of the qubits involved in the computation state becomes corrupted, the success probability of the entire computation is decreased. Trivially, the circuit coherence is bounded above by the product of circuit depth and circuit width, since it may be the case that every qubit is in the computation step at every timestep. This is not always the case, per our discussion of reusable ancillae above, and in fact, it is our goal to show that there is an asymptotic separation between the circuit coherence and the circuit depth width product for factoring.