

Design for a Molecule-Based Quantum Processor

A detailed blueprint for a molecular-spin-based quantum processor brings this technology closer to reality.

By **Ryan Wilkinson**

Established quantum-computing platforms, such as superconducting circuits and ion traps, suffer from problems related to error mitigation and correction. These issues could be avoided by using a quantum processor in which the usual two-level computational units (qubits) are replaced with multilevel units (qudits). Now Stefano Carretta at the University of Parma in Italy and his colleagues propose a design for such a processor that uses qudits made from molecular spins [1]. The researchers lay out methods to implement elementary qudit operations in their processor and to read out the computational output.

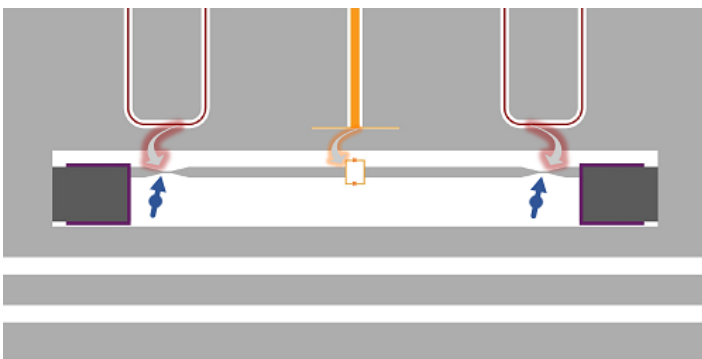
The basic building block of the new processor contains two molecular-spin qudits coupled to a superconducting resonator. The resonator reads out the states of the qudits and mediates an interaction between them that allows the execution of quantum operations. So-called transmission lines generate local electromagnetic pulses that enable the control of each qudit's state.

Carretta and his colleagues demonstrate that their scheme can be generalized to an array of such components. They then show how to encode several qubits within a single molecular-spin qudit, providing a route to simplifying quantum operations, which could limit errors. The researchers also present schemes for executing operations involving one or two qudits, which they test in simulations based on a realistic experimental layout. According to the team, the successful outcomes of these tests suggest that this platform could provide the basis for a universal quantum processor, the “holy grail” of quantum computing.

Ryan Wilkinson is a Corresponding Editor for *Physics Magazine* based in Durham, UK.

REFERENCES

1. A. Chiesa *et al.*, “Blueprint for a molecular-spin quantum processor,” *Phys. Rev. Appl.* **19**, 064060 (2023).



Credit: A. Chiesa *et al.* [1]