An Unconventional Computing Approach for Quantum Computer Control and Quantum Memory

Technical Document ZF003 v1.01, July 27, 2020

Erik P. DeBenedictis
Zettaflops, LLC, Albuquerque, NM 87112
erikdebenedictis@zettaflops.org

Overview

This document proposes an "unconventional computing" technology for controlling quantum computers, including specific application to quantum memory based on posits. While improved qubits are a top priority in quantum computing, existing approaches for classical control systems limit scale up to 50-1,000 qubits due to wiring between the cryogenic environment and room temperature[1] or excessive (CV^2) dissipation in cryogenic electronics.[2]

We introduce a novel approach for the cryogenic portion of the classical control system. The approach exploits unique properties of the cryogenic environment to reduce dissipation per function by at least $10^3\times$, thereby permitting additional scale up of $10^3\times$. This would enable demonstrations of quantum error correction, the quantum equivalent of floating point, and a subroutine capability that would make much more sophisticated algorithms feasible.

This document also highlights a quantum number system like posits. While qubits and their rotations are analog, qubit values will be digital while in source code and in the classical control system. There are situations where the necessary precision varies with value, such as the rotations in a QFT, so we anticipate the need for a floating-point like "quantum posit[3]" format and a digital-to-qubit conversion device[4] to load values into qubits and quantum memories.

In the longer term, these capabilities could make sophisticated quantum algorithms that require error correction practical, such as quantum linear algebra and resulting applications in engineering and machine learning.

Background

A 53-transmon quantum computer[1] recently showed performance exceeding the world's largest supercomputer, yet it was close to its scaling limit due to wire congestion and resulting heat back flow. Advocates of spin qubits implied they had a solution scalable to 10^8 qubits,[5, 6] but when researchers fabricated a CMOS classical control system for spin qubits and extrapolated scaling limits based on CMOS's CV^2 power dissipation,[2] their graph only went up to 10^3 qubits.

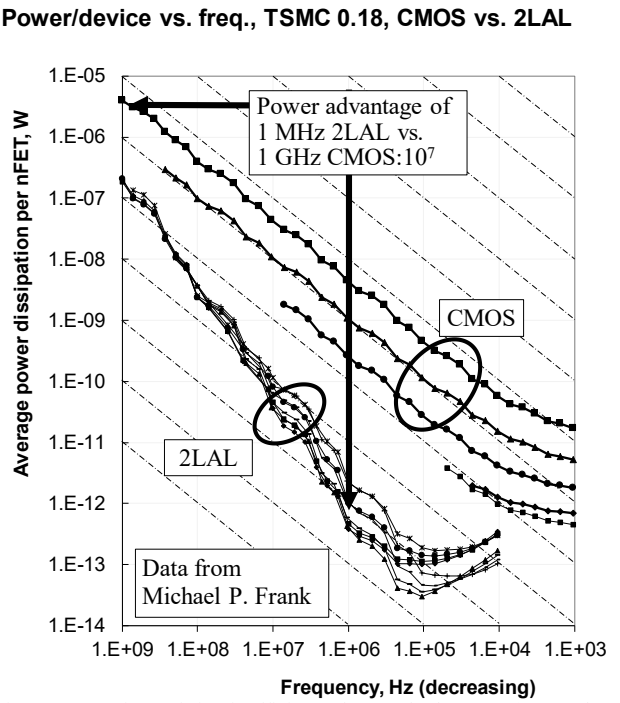

Fig. 1. Comparison of circuit efficiency for standard CMOS (top) and an adiabatic circuit 2LAL (bottom), showing a maximum advantage of 1,000× at 200 KHz. However, if 2LAL is operated at 4 K, down sloping curves should extend further, leading to a possible 100,000× energy efficiency improvement over room-temperature electronics. This may allow transistorized 2LAL to compete with JJs in applications where speed is not essential.

As shown in fig. 1, reversible logic[7] uses a different circuit design that should reduce power per function by 10^7 and energy by around 10^3 compared to CMOS. This document shows how to practically exploit the heretofore elusive benefits of reversible computing.

While a microwave signal can be generated at room temperature, routed to the cryogenic environment over a single wire, and execute a quantum operation on many qubits, other operations require a loop governed by a quantum measurement. If wires to room temperature are considered a scaling bottleneck, the classical electronics for these operations must be in the cryogenic environment, making energy efficiency improvements critical.

Linear algebra is currently the domain of classical supercomputers for applications such as simulations of structures via the finite element method. The original quantum linear algebra algorithm is called HHL,[8] solving the matrix equation $Ax = b$, where A is a matrix, b is an input vector, and x is the result. There are currently implementations of the HHL algorithm on existing small-scale quantum computers (Qiskit[9]). IARPA's QCS program did a study of resource requirements for radar scattering analysis on a quantum computer.[10] Quantum linear algebra, simulation, and optimization applications are all potential long-term direction for the ideas in this document.

Unconventional classical control system

Fig. 2 illustrates the unconventional technology, documented in greater detail in ref. 11. Just as CMOS and DRAM comprise a ubiquitous technology hybrid for room-temperature computing, a hybrid of cryogenic adiabatic (reversible) transistor circuits (CATC) and SFQ-type Josephson junction (JJ) electronics form an effective hybrid for the purposes of this document. Transistors

are small, hence dense, but slow when run adiabatically, while JJs are fast but large. The combination illustrated by the layers in fig. 2b-c meet the entire range of requirements for a classical control system, as explained further below.

Architecturally, we find JJs suitable for fast reconfigurable logic (fig. 2a) and microwave switches (fig. 2b, right). However, the small size of transistors makes them ideal for memory-like functions, including storing multiple FPGA configuration strings, storage of microwave envelopes (waveforms), and floating point (posit) data.

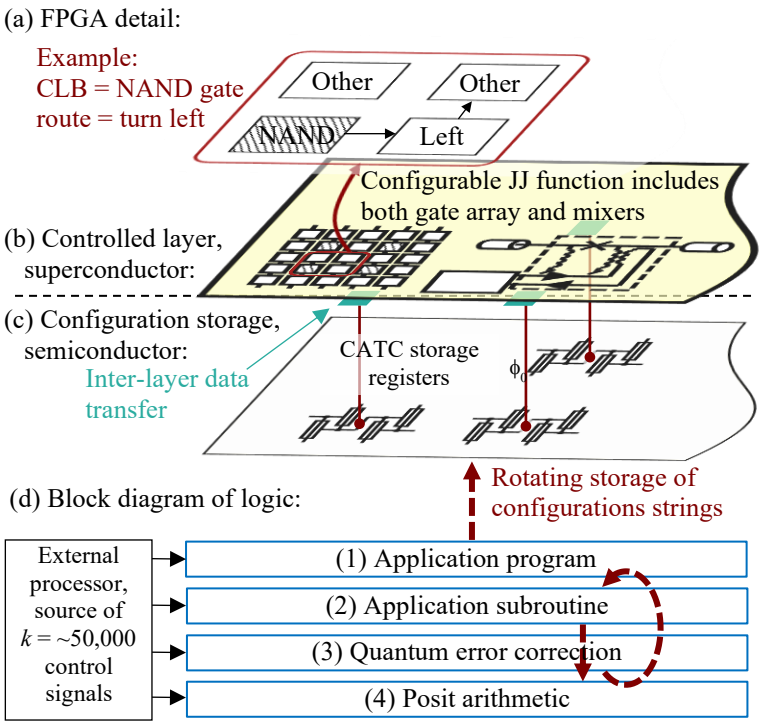

Fig. 2. Scalable classical control system. (a) Detail of FPGA block with a controlled logic block and routers. (b) JJ-based controlled layer including FPGA on the left and a microwave mixer on the right. JJs are physically large, so the density is low. (c) Configuration storage layer using dense, low-power CATC shift registers, which are slow. (d) Logical block diagram: External processor loads 4 × k-bit configuration strings (1)-(4) that can reprogram the FPGA layer with one parallel shift.

Fig. 2 illustrates architectural issues related to latency and throughput. A microprocessor must have fast conditional branches in scalar code or throughput will suffer. However, a quantum computer will get its throughput mostly from quantum speedup resulting from entangled qubits. The unconventional approach being proposed here uses slower and hence lower power electronics for reconfiguring what is essentially an FPGA in ~1μs while using a smaller number of faster devices where needed for fast loops and switching microwaves.

Fig. 2d shows the system-level behavior, with the reconfiguration from shifting between configuration strings (1-4) equivalent to a subroutine. The illustration shows (1) an application program calling (2) a subroutine within the application. The subroutine might call (2) quantum error correction in the event of an unexpected error or (4) quantum arithmetic as another level of a subroutine within the application. The enabling concept is that adiabatic transistors are physically small so they can contain a lot of state but slow. While slow, the adiabatic transistors switch in less than the decoherence time of qubits, so the system is fast enough for its intended purpose without being so fast that heat dissipation is excessive. For detail see ref. 11.

The general approach in this document is to engineer the electronics at the intersection of classical and quantum information to its unique environment. Irrespective of the JJs, transistors, and reversible circuits that were used in the example above, the idea is to exploit diverse opportunities offered by the cryogenic environment whenever possible. We feel free to set aside

the architectural constraints of microprocessor design because we are creating a quantum computer controller—and besides, there could very well be microprocessors at the room temperature stage.

Classical-quantum autonomous functions enable quantum error correction and quantum memory

Fig. 3 illustrates a potential test setup to show the benefits of the unconventional computing approach for quantum computer control. The goal is to perform certain classical-quantum functions autonomously (i. e. without a lot of wires to room temperature) and at 1,000× lower power than current methods and hence more scalable.

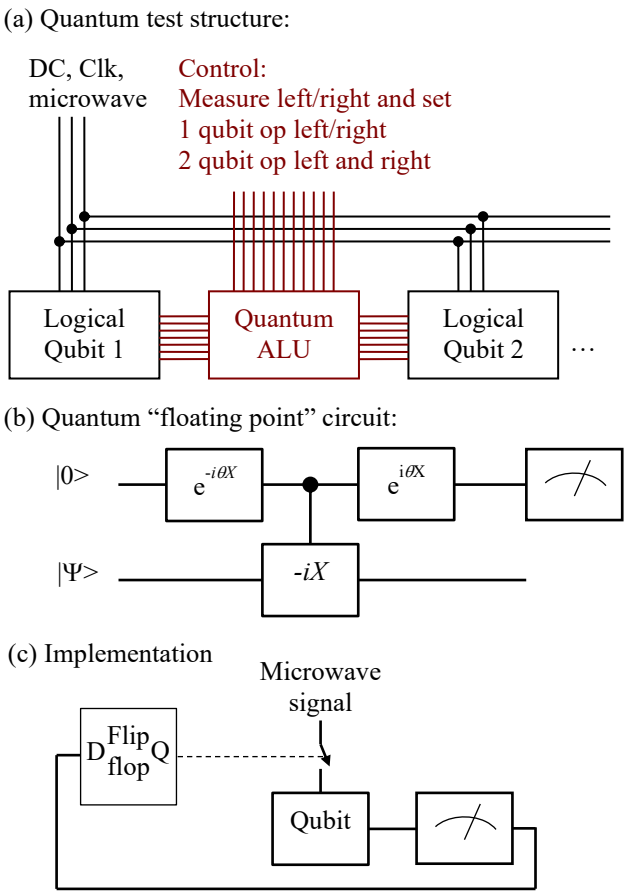

Fig. 3. (a) Quantum control test structure; logical qubits share control signals, but error correction behavior is local. Test harness or posit arithmetic in red. (b) Exemplary preparation for a small angle and (c) implementation as a mixed classical-quantum hybrid circuit to generate them. See ref. 4 for more information on (b) and (c).

The top-level block diagram is shown in fig. 3a. The black components in the diagram show a set of room temperature signals (DC, Clk, microwave signal) routed to multiple logical qubits. Each qubit would detect and correct errors without separate wires to room temperature. Fig. 3a also includes a quantum ALU in red, which could operate on the logical qubit, initially for testing but eventually for the execution of real algorithms.

Fig. 3b is an illustration of a classical-quantum loop, which could be used for quantum error correction, but the circuit shown is part of "quantum floating point.[4]" Fig. 3c shows a simplified implementation: A classical flip flop (using CATC) would hold system state, controlling a microwave switch such as in fig. 2b. If enabled, the microwave signal would affect a qubit, which is measured. The measurement feeds back to the flip flop and affects system state.

Quantum error correction

The energy efficiency advantages in the computational substrate illustrated in figs. 1-2 could make an important step forward in quantum error correction within a five-year timeframe. The original work on quantum error correction included 5-bit, 7-bit, and other simple codes. It

should be possible to use the structure shown in fig. 3 to test in situ, autonomous control of logical qubits, proving the result by measuring an increase in decoherence time.

Quantum posit memory

The input to the HHL algorithm,[8] quantum simulation, and quantum optimization comprise matrices A or H and vector *b* as superpositions of numeric data encoded into the state of a qubit register. The current expectation is that the values $|A_{ij}\rangle$ and $|b_j\rangle$ would be created by using a microwave waveform generated at room temperature to rotate a $|0\rangle$ qubit. The pulse length determines the amount of rotation and hence the value to be left in the qubit and subsequently stored in the superposition state or quantum memory. This approach requires a microwave cable from room temperature for each qubit, which impairs scalability.

The alternative approach in fig. 4 is to create the equivalent of a floating-point number system for quantum computers. The quantum computer would be able to perform Clifford operations, etc., as usual. However, it would also store posit-encoded θ_i values in cryogenic classical memory and could perform $R_X(\theta_i)$ on a $|0\rangle$ qubit to create analog qubit values. The resulting rotated qubits would be put into a quantum memory and form the superposition state of A, H, or *b*.

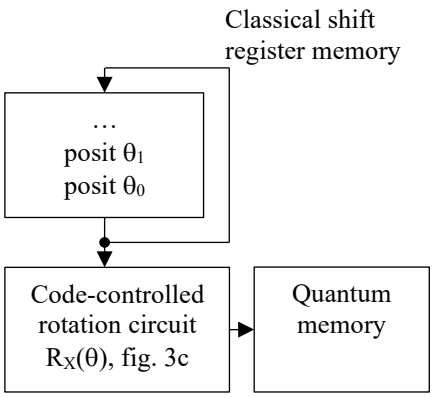

Fig. 4. Classical memory as an energy efficient *k*-bit shift register feeds a posit-to-rotation converter,[4] and the resulting qubit is added to a quantum memory.

Potential future work

Taking the classical control system in fig. 2 as the "unconventional computer," the expectation is that the structure would be in a cryogenic environment with cables or optical fibers connecting it to room temperature electronics. Some of the room temperature electronics would comprise programmable signal generators for DC, AC, or microwave waveforms as indicated in fig. 3a. Room temperature FPGAs and/or microprocessors would supply data to the signal generators or send data to the structure in fig. 2 to (a) load programmable structures or memory and (b) to send or receive data from the qubits. Loading the classical memories would only occur between program runs when timing is not critical.

The ideas in this document should work with a quantum computer "intermediate representation," once one is developed. Say an application like radar scattering[10] is to run on the quantum computer. That application is too big for present tools, but QisKit, Q#, etc. perform the right type of processing and could be scaled up. The output of these tools includes information for classical simulators (for debugging) and back ends for various quantum computers. A back end for the structure in fig. 2 would need to be created.

Data defining a real-world problem and stored in the posit-based quantum memory will be in the form of a series of floating-point parameters, presumably in text or IEEE format. However, these number formats translate easily to both the proposed "quantum posit" bit encoding and

equivalent reversible arithmetic circuits,[12] allowing for different precision and range. Part of this document is a system that would load classical, bit encoded, posit data into qubits.

While not explained in previous sections, a control system of the type described should be able to do interesting types of readout—such as state tomography. Fig. 2d shows schematically how the control system could load and run a quantum program. Given this, it would be easy enough to run a quantum program repetitively. Quantum measurements are single bits (unfortunately), but the classical control system could run a program repetitively, compute statistics on the output bits, converting them into numbers, even posits. The posits could be sent to room temperature and printed.

At the hardware level, SFQ-type Josephson junction chips are generally fabricated on blank silicon wafers by organizations like Lincoln Laboratories, SkyWater, and SeeQC. We know at least one instance where a fab has fabricated a JJ chip on a non-blank silicon wafer. IARPA has a SuperTools program where Synopsys is developing relevant design tools. These existing technologies would apply to the top layer in fig. 2

The proposal in Ref. 11 involves creating a hybrid transistor-SFQ-type JJ FPGA. While FPGA tools are mature, there are no current tools with a back end that would support some of the unique properties arising from the system proposed.

Silicon design tools can simulate and create masks for adiabatic and reversible transistor circuits, but the design tools are not cognizant of the special adiabatic and reversible properties of the circuit families.

References

[1] Arute, Frank, et al. "Quantum supremacy using a programmable superconducting processor." *Nature* 574.7779 (2019): 505-510.

[2] Pauka, S. J., et al. "A cryogenic interface for controlling many qubits." *arXiv preprint arXiv:*1912.01299 (2019) Note see fig. 4e p. 6.

[3] Gustafson, John L., and Isaac T. Yonemoto. "Beating floating point at its own game: Posit arithmetic." *Supercomputing Frontiers and Innovations* 4.2 (2017): 71-86.

[4] Wiebe, Nathan, and Vadym Kliuchnikov. "Floating point representations in quantum circuit synthesis." *New Journal of Physics* 15.9 (2013): 093041.

[5] Veldhorst, M., et al. "Silicon CMOS architecture for a spin-based quantum computer." *Nature communications* 8.1 (2017): 1766. Note phrase on p. 6: "A detailed analysis based on a specific design and targeted operation, going beyond this work, will therefore be highly valuable."

[6] Conway Lamb, Ian. *Cryogenic Control Beyond 100 Qubits*. MS thesis. University of Sydney, 2016.

[7] Frank, Michael Patrick, and Thomas F. Knight Jr. *Reversibility for efficient computing*. Diss. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1999.

[8] Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations." *Physical review letters* 103.15 (2009): 150502.

[9] IBM corporation, *Solving Linear Systems of Equations using HHL*. https://qiskit.org/textbook/ch-applications/hhl_tutorial.html.

[10] Scherer, Artur, et al. "Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target." *Quantum Information Processing* 16.3 (2017): 60.

[11] DeBenedictis, Erik P. "Quantum Computer Control using Novel, Hybrid Semiconductor-Superconductor Electronics." *arXiv preprint arXiv:*1912.11532 (2019).

[12] Soeken, Mathias, et al. "Logic synthesis for quantum computing." *arXiv preprint arXiv:*1706.02721 (2017).