Architectural Considerations in the Design of a Superconducting Quantum Annealing Processor

Paul I. Bunyk, *Member, IEEE*, Emile M. Hoskinson, Mark W. Johnson, *Member, IEEE*, Elena Tolkacheva,
Fabio Altomare, Andrew J. Berkley, Richard Harris, Jeremy P. Hilton, Trevor Lanting,
Anthony J. Przybysz, and Jed Whittaker

(Invited Paper)

Abstract—We have developed a quantum annealing processor, based on an array of tunable coupled rf-SQUID flux qubits, fabricated in a superconducting integrated circuit process. Implementing this type of processor at a scale of 512 qubits and 1472 programmable interqubit couplers and operating at ∼20 mK has required attention to a number of considerations that one may ignore at the smaller scale of a few dozen or so devices. Here, we discuss some of these considerations, and the delicate balance necessary for the construction of a practical processor that respects the demanding physical requirements imposed by a quantum algorithm. In particular, we will review some of the design tradeoffs at play in the floor planning of the physical layout, driven by the desire to have an algorithmically useful set of interqubit couplers, and the simultaneous need to embed programmable control circuitry into the processor fabric. In this context, we have developed a new ultralow-power embedded superconducting digital-to-analog flux converter (DAC) used to program the processor with zero static power dissipation, optimized to achieve maximum flux storage density per unit area. The 512 single-stage, 3520 two-stage, and 512 three-stage flux DACs are controlled with an XYZ addressing scheme requiring 56 wires. Our estimate of on-chip dissipated energy for worst-case reprogramming of the whole processor is ∼65 fJ. Several chips based on this architecture have been fabricated and operated successfully at our facility, as well as two outside facilities (see, for example, the recent reporting by Jones).

Index Terms—Computational physics, quantum computing, superconducting integrated circuits.

I. INTRODUCTION

PROPOSED implementations of quantum computers capable of solving problems at a useful scale generally involve at least many thousands of qubits. Whether the algorithm envisioned is based on the quantum circuit model, or on an adiabatic method, there are a number of physical requirements that constrain the design of a large-scale manufactured quantum device. The device architecture must facilitate precise individual qubit control, computationally interesting interaction between qubits,

and high fidelity readout of qubit state. Of particular importance are the practical constraints that arise in achieving these design goals while maintaining the carefully engineered environment required to implement a quantum algorithm.

One of the advantages of an approach based on superconducting qubits is that a largely compatible classical electronics technology is known and available in the guise of single-flux-quanta-based circuit architectures. The authors have previously presented the architecture, design, and operation of an SFQ-based system for controlling [3] and reading out [4] a quantum annealing processor based on flux qubits [1], [5] at the core of D-Wave One system. The authors, as well as a number of other researchers, have gained some experience using this first generation processor [6]–[9]. This experience has informed and guided the design of a second generation quantum annealing processor, the D-Wave Two system. In this paper, we provide an overview of this processor's architecture and discuss some of the considerations and tradeoffs involved in its design.

At the root of the design problem is the operation of a quantum annealing processor based on superconducting flux qubits. Many of the processor building blocks are described by Harris *et al.* [1], [10]. A number of the control terminals (biases) required by the qubits and interqubit couplers are discussed by Johnson *et al.* [3], and these are largely unchanged. The implementation of the control circuitry, the cross section of the fabrication process, and the number of devices used are among the main differences between the two generations of D-Wave processors.

The D-Wave One used current biased SFQ [11], [12] demultiplexer circuitry to address all N programmable devices on chip, requiring an asymptotically optimal $O(\log(N))$ number of control lines. This circuitry was designed with consideration for the need to minimize static power dissipated on chip during programming, and to this end used very low bias supply voltages, and very low value shunt resistors. The predicted peak temperature of the junction shunts during programming, about 500 mK, was sufficiently low to ensure negligible thermally induced bit errors in digital-to-analog converter (DAC) programming.

We found, however, that it took of order 1 s for this heat to dissipate sufficiently for the processor to return to ∼20 mK, a temperature low enough to run the quantum annealing

algorithm and obtain solutions to posed problems with appreciable probability. As typical computation time is $\sim 20\ \mu$s, this was clearly an unacceptable amount of time to wait to run the algorithm after programming.

D-Wave Two control circuitry was designed to eliminate static power dissipation completely and to simplify the design greatly with an XYZ addressing scheme requiring $O(\sqrt[3]{N})$ lines. Though not logarithmic, this scaling is sufficiently weak to allow processors significantly larger than the one under discussion to be operated in our existing apparatus.

Improvement in performance of the annealing algorithm was also a central design goal of the D-Wave Two. In a fixed temperature environment, performance can be improved by increasing the qubit energy scale. This can be done by decreasing qubit inductance and capacitance. Given constraints on dielectric permittivity and wiring geometry, this is most practically accomplished in our design by reducing the physical length of the qubit wiring. Reduction of length by a factor of two compared with D-Wave One was achieved by adding two metal layers to the fabrication process, for a total of 6 superconducting metal layers. This allowed for an increase of overall processor density by a factor of four.

In the first section of this paper we will step back and discuss the requirements of the whole chip and give our rationale behind the chosen hardware graph topology (common between D-Wave One and Two) from the top-down perspective. We will then be in a position to present the chosen implementation of D-Wave Two control circuitry in bottom-up fashion. Read-out infrastructure will be described in a subsequent publication.

II. CIRCUIT TOPOLOGY

D-Wave quantum annealing processors have evolved through a series of generations subject to the competing pressures exerted by computational complexity and practical implementation. They are designed to solve this problem: given *hardware graph G*, minimize the following quadratic form over discrete variables $s_i \in \{-1, +1\}$:

$$E(\vec{\mathbf{s}}|\vec{\mathbf{h}}, \hat{\mathbf{J}}) = \sum_{i \in \text{nodes(G)}} h_i s_i + \sum_{\langle i,j \rangle \in \text{edges(G)}} J_{ij} s_i s_j$$

with problem parameters $h_i, J_{ij} \in \{-1, -7/8, \ldots + 7/8, +1\}$ for our current hardware.

Our current hardware graph topology, which we named Chimera, was designed to satisfy a number of common-sense requirements related to its intended use for solving optimization problems, subject to a number of physical implementation constraints.

A. Requirements

1) **Non-planarity**: We desire to tackle NP-complete problems, so non-planarity of the underlying graph is an important condition to making the corresponding Ising-spin problem NP-complete [13], [14]. A related motivation is that non-planarity is required to establish chains of qubits that cross each other.

2) **The ability to embed complete graphs**: To solve Ising spin glass problems with different topologies using a single processor, it must be possible to map, or *embed* the problem graph into the available hardware graph. This process typically involves using chains, trees, or other connected sub-components of the hardware graph (comprising *physical qubits*, strongly ferromagnetically coupled to each other) to represent a single node in the problem graph (*logical qubit*). In the language of graph theory, we want the hardware graph to have largest possible variety of problem graphs as its *minors* [15]. While embedding arbitrary graphs is computationally hard, one can nonetheless determine how large of a degree M complete graph K_M can be embedded in a given size hardware graph, thus guaranteeing that all graphs up to M nodes are embeddable using a straightforward prescription.

3) **The ability to incorporate on-chip control circuitry**: While a single qubit, or a handful of them, can be precisely controlled with dedicated analog lines driven by room-temperature electronics, integrating more than a few dozen qubits on a single chip requires some on-chip control circuitry. For example, there are currently six control "knobs" for every qubit, required to make the qubits robust to fabrication variability [10], and one "knob" per coupler. Where possible, we have designed our QA processors to use *static* flux biases applied to target superconducting loops in order to realize most of these knobs. The desired values of flux biases are programmed into individual control devices using a relatively small number of essentially digital control lines that carry signals generated at room temperature. These control devices combine the functions of persistent memory and digital-to-analog conversion. We call these devices flux DACs, or Φ-DACs. From an architectural viewpoint, each Φ-DAC is a relatively macroscopic object with a typical size of $\sim 10\ \mu$m. Having several of them attached to a single qubit sets a lower bound on qubit size and influences possible qubit shapes and hardware graph topologies.

B. Constraints

1) **Limited qubit fan-out**: From an applications standpoint, the best option would be for each qubit to be connected to all others. However, directly implementing a complete K_M graph in hardware for an arbitrarily large M is impractical. Each qubit in our current design can be connected to only a relatively small number of other qubits $\lesssim 10$ before non-ideal features arise in the qubit response and the coupling energy scale (compared to $k_b T$, for example), becomes too small.

2) **Minimizing uncoupled qubit/coupler lengths**: To optimize qubit energy scales and coupling strengths, neither qubits nor couplers can be designed to span an arbitrary length (e.g., full size of a large processor matrix). Ideally, the entire qubit length should be magnetically coupled to connected couplers, and the entire coupler length to connected qubits.

3) **Minimization of noise pickup and crosstalks**: Flux qubits and couplers are rf SQUIDs, which can be quite sensitive to magnetic fields. Extreme care needs to be taken to minimize their pickup of undesired disturbances, such as coupling to external flux noise sources or the unintended coupling (**crosstalk**) from a control line to a device it is not intended to control.

4) **2D chip integration**: While it would be nice to be able to grow a processor lattice in all three dimensions, in reality these lattices have to be implemented on the surface of 2D chips. Even if we imagine adding more metal layers to our fabrication process or 3D integration of several chips stacked on top of each other and passing quantum states between them (through, e.g., superconducting backside vias), growing the processor graph along the *physical* third dimension will always be harder than along the 2D chip plane.[1]

5) **Regularity and the notion of a "unit tile"**: While it is in principle possible to arrange qubits in highly irregular structures, in practice, especially while designing chips not tailored to any specific problem graph structures which might arise in a concrete application, we find it convenient (to simplify the design and operation) to introduce the notion of a *unit tile*, which is a smaller structure that can be replicated in both dimensions of the chip plane.

C. Chimera Topology

One of the features of flux qubits is that (unlike, e.g., qubits based on quantum dots or individual trapped ions) they are essentially macroscopic inductive loops interrupted by Josephson junction(s) and that these qubit body loops can be stretched and routed as needed. The same is true for qubit-to-qubit couplers [16]–[18], except that parametrically they tend to be lower inductance devices, and thus, shorter.

With this in mind we examined different arrangements of qubit loops, and eventually settled on the Chimera unit tile topology (used in both D-Wave One and D-Wave Two processors), schematically depicted in Fig. 1. Each unit tile consists of 8 qubits—4 horizontal and 4 vertical—with couplers between each horizontal/vertical pair. The unit tile is a complete bipartite graph $K_{4,4}$. Unit tiles can be arranged into larger grid-like structures that fill a plane, and each horizontal qubit can be coupled to the corresponding qubits in the neighboring tiles to the left and right, while each vertical qubit can be coupled to those in the neighboring tiles above and below.

How well does the Chimera topology satisfy the requirements and constraints given above? Consider the following:

• The Chimera graph is non-planar. Assuming the ability to establish chains of qubits along rows and columns of the processor matrix, there is a straightforward approach to embed complete graphs up to $4N$ nodes in a $N \times N$ grid of unit cells (denoted as C_N). This is illustrated in Fig. 2

[1]Of course, one can attempt embedding *logically* higher-dimensional structures into planar chips, but this approach soon runs afoul of the second constraint above.
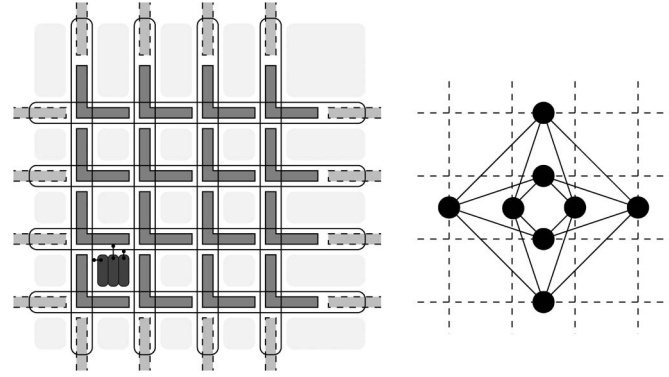

Fig. 1. Chimera unit cell topology. (Left) Layout sketch: Qubit bodies are represented by the elongated loops that span the whole width/height of the unit tile. Each qubit is coupled to four others within the unit tile via the *internal coupler* bodies (dark L-shaped objects). Qubits are coupled to others in neighboring tiles via *external couplers* (lighter dashed rectangles). Control circuitry (Φ-DACs and corresponding analog control structures) are placed within light-shaded areas between the qubit/coupler bodies. (Right) Graph representation: Each unit tile corresponds to a complete bipartite graph $K_{4,4}$ (dark nodes and solid line edges). Qubits from different tiles are coupled in square grid fashion (dashed edges).

for the case of $N = 4$. This approach can be validated based on the following observations:

1) Taking a single $K_{M,M}$ tile and ferromagnetically coupling pairs of horizontal and vertical qubits along its diagonal (*contracting edges*, which connect them in graph-theoretical language) produces a complete graph K_M.

2) Taking a 2×2 array of complete bipartite graphs $K_{M,M}$ and ferromagnetically coupling pairs of qubits in the same row/column produces a complete bipartite graph $K_{2M,2M}$.

3) Taking two complete graphs K_M and connecting them to two sides of a complete bipartite graph $K_{M,M}$ produces a complete graph K_{2M}—every node in K_{2M} can be coupled to every other node either because they either belong to the same K_M and were coupled anyway or because they belong to different K_Ms, in which case there is connection between them through the complete bipartite part.

• The Chimera topology was designed to be interleaved with the required control circuitry (which is schematically represented by lighter shaded areas in Fig. 1). In the implementation under discussion, each square "plaquette" formed at the intersection of two qubits contains three Φ-DACs. Generally, the left (right) Φ-DAC provides certain type of control to the vertical (horizontal) qubit, while the middle one controls the corresponding coupler, as schematically shown in the bottom-left corner of this diagram.

• Almost all of the qubit length is coupled to couplers, and almost all of the coupler length is coupled to qubits, thus maximizing coupled signal strength. Also, implementing the qubit and coupler loops as long and narrow differential microstrip lines (in practice, over a superconducting ground-plane) minimizes noise and parasitic cross-talk pick-up.
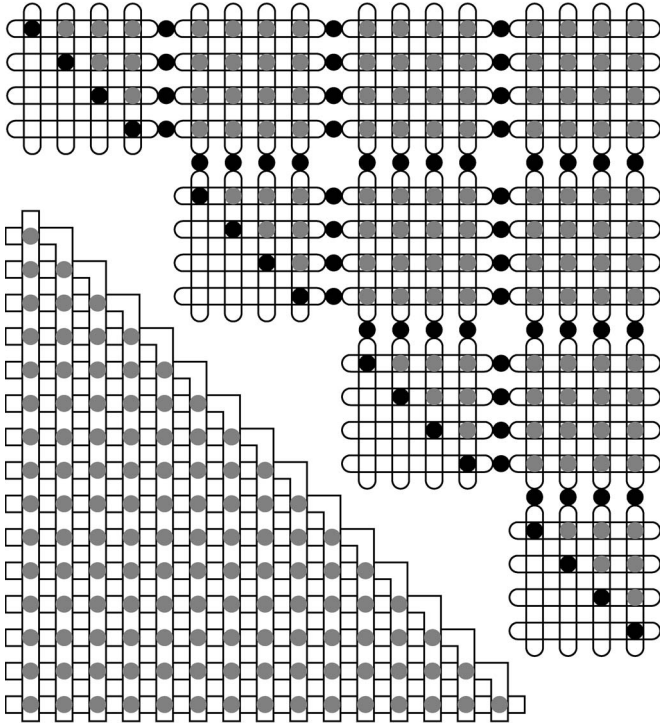
Fig. 2. Embedding complete graphs in the Chimera topology. (Top) Strong ferromagnetic couplings (black circles) along the diagonal of the C_N matrix ($N = 4$ in this example), as well as along rows and columns between tiles in half of a C_N (upper left triangle shown) connects 80 physical qubits into 16 chains. (Bottom) Each chain can be coupled to every other chain via sign and magnitude tunable couplers (gray circles), thus embedding a complete graph ($K_{4N=16}$ in this example).

- Chimera unit tiles can be arranged into arbitrarily large 2D structures (limited only by fabrication yields, die size and available number of IO lines/die pads required to program all Φ-DACs). For example, the D-Wave One processor contained 128 qubits in a C_4 grid (a 4×4 grid of 8-qubit tiles) and the D-Wave Two processor contains 512 qubits in a C_8 grid (an 8×8 grid of 8-qubit tiles).

While this approach can be generalized to an arbitrary $K_{M,M}$ unit tile with $2M$ qubits, our current implementation of $M = 4$ was chosen because (as will be seen later) all of its required Φ-DACs can be fit in a 5×5 array of fixed-size plaquettes without too much wasted space, simplifying (manual) layout and managing overall design complexity. Another advantageous feature of the $M = 4$ size is that the number of Φ-DACs used for problem specification (one per qubit and one per coupler, thus giving a total of 32 per tile) is approximately balanced with the number of Φ-DACs used to make qubits robust against fabrication variations (4 per qubit in the D-Wave One generation, 5 in D-Wave Two, giving a total of 32 or 40 per tile, respectively). For smaller unit tile size a majority of the Φ-DACs would be of the second variety.

III. DESIGN AND OPERATION OF A Φ-DAC

The precision desired for setting problem parameters sets the requirements for the range and precision of individual Φ-DACs. Generally, our current implementation requires about 8 bits of dynamic range for individual Φ-DACs, with full ranges varying

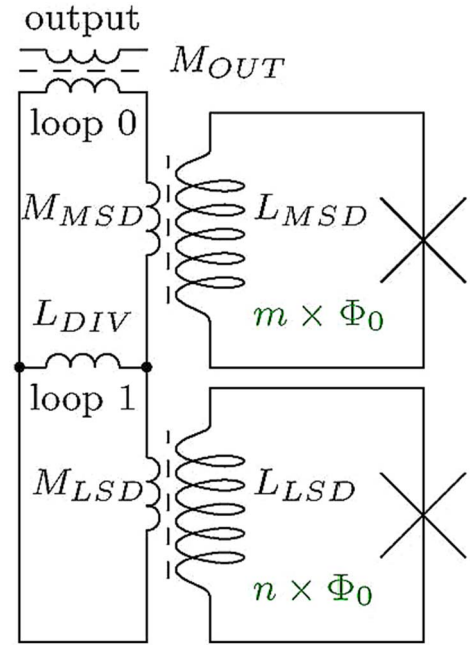Fig. 3. Schematic view of generic two-stage D-Wave Two Φ-DAC.

from several thousandths of a magnetic flux quantum ($m\Phi_0$) to half a Φ_0 coupled into qubit or coupler control loops, depending on the Φ-DAC type.

To achieve this dynamic range while minimizing both total area occupied by control circuitry (thus minimizing qubit length and increasing qubit energy scales) and total number of wires needed for programming, in our current design we chose to implement most of our Φ-DACs as two-stage devices, of the kind schematically shown in Fig. 3.

Each of the DAC digits (referred to as "most significant" and "least significant" here, or "MSD" and "LSD") is implemented as a SQUID loop into which we can write and store some number of flux quanta m, e.g., $-8 \lesssim m \lesssim 8$. Individual quanta can be added to or subtracted from the storage loop via an SFQ pulse source, depicted here as a Josephson junction; its structure and operation will be described in Section III-B. Both digit storage loops are magnetically coupled into an output device via an inductive ladder.

A single flux quantum added to the MSD coil induces $(M_{\mathrm{MSD}}/L_{\mathrm{MSD}}) \times \Phi_0$ flux into the top ladder loop, and $(M_{\mathrm{MSD}}/L_{\mathrm{MSD}} \times M_{\mathrm{OUT}}/L_{\mathrm{loop0}}^{total} \times \Phi_0$ into the output device ($L_{\mathrm{loop0}}^{total}$ denotes total inductance of the MSD loop). The output flux increases proportionally with the number of flux quanta added, up to a maximum determined by the device parameters and addressing scheme. There is in general a nonlinear component associated with the junction inductances, but as long as these inductances are small compared to main loop inductances (true for our devices), this correction is negligible. In our example, if MSD loop can store up to 8 flux quanta of either polarity, it can provide 16 distinct values of output flux, or implement a 4-bit DAC.

To increase precision, a second stage is added. Here, the effect of a single flux quantum in the LSD loop is further subdivided by a factor of $L_{\mathrm{DIV}}/L_{\mathrm{loop1}}^{total}$. If this loop can also
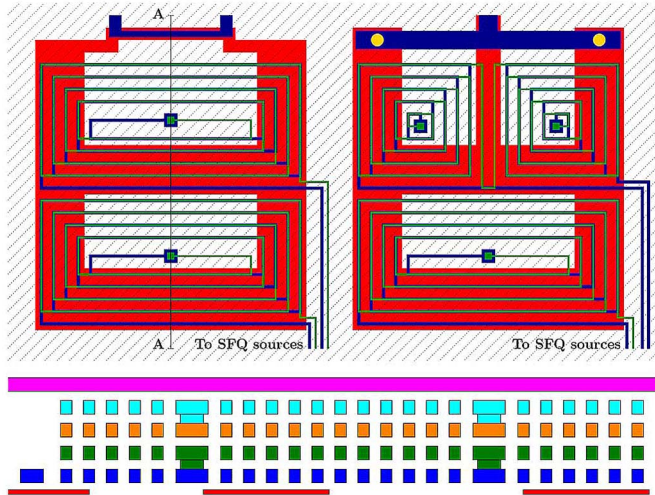
Fig. 4. (Top) Simplified layout of two typical Φ-DACs. (Left) Low range, magnetically coupled to the output, and (right) high-range (e.g., coupler Φ-DAC), merging device to be controlled (a compound Josephson junction) with MSD ladder stage. Leads of the coils connect to SFQ pulse sources (not shown). (Bottom) CAD view of a cross-section of a real Φ-DAC layout (to scale; width and spacing of spiral wires are 0.25 μm) along cut line similar to line A-A shown on the low-range (left) device.

provide 16 distinct values of stored flux and the division ratio is 16, one MSD step will be further subdivided into 16 steps of LSD, and the two-stage device is an 8-bit DAC.

In practice, of course, we want to guarantee both the total output range and the coverage of an MSD step by the LSD in the presence of fabrication variations, so we need to add some margin to the number of quanta that we can store in both loops.

Φ-DACs with different numbers of digits and weights of each digit can be designed using the same principles, but we have found that this two-stage design is sufficient for almost all of our DACs.[2]

A. Φ-DAC: Inductive Storage and Ladder

Having covered the basic idea behind our Φ-DACs, we can present a more realistic layout of their implementation as shown by the top portion of Fig. 4. Large storage inductors ($\sim$1 nH) are implemented as stacked spirals (blue and green in the figure), shown here wound in two metal layers, though our real layouts use four layer spirals, with 0.25 μm line width and spacing design rules.

The inductive ladder is implemented as two galvanically connected washers in the bottom metal layer (red), magnetically coupled to their two coils. The horizontal bar between them implements the shared inductance L_{DIV} of Fig. 3.

To minimize unintended coupling between DAC coils and other elements of the circuit, the whole structure is covered by a shielding sky-plane in the top metal layer (dotted diagonal lines).

[2]Two special cases were introduced in D-Wave Two processors: a DAC which biases the qubit compound Josephson junction major loop, for which currently 5 bits of dynamic range is sufficient and it was implemented as a single coil of the same type directly coupled into a target device, and second a very coarse stage for a qubit flux bias DAC, useful to deal with larger local qubit flux offsets.

Simple magnetic coupling between the inductive ladder and target device using the microstrip transformer shown in the top-left panel of Fig. 4 is sufficient to implement a full range of several tens of mΦ_0 into the target device. However, the majority of our DACs, ones that control the compound Josephson junctions (CJJs) of couplers, inductance tuners and persistent current compensators, require a range comparable to half a Φ_0, since they need to be able to bias their target's corresponding CJJ all the way between its maximum I_c and fully suppressed. To implement such higher-range control we merge the CJJ loop of the target device with the MSD stage of the inductive ladder, as shown in the top-right panel of Fig. 4 (Josephson junctions are shown as yellow circles).

An additional complication of this particular structure is that the DAC should be coupled with equal strength into both halves of the target CJJ loop in order to avoid coupling into target device body. To achieve this, the MSD coil is split into two symmetric halves, as shown in the top-right panel of Fig. 4.

The simplistic lumped-element model of Fig. 3 is not entirely adequate as a complete description of our Φ-DAC devices, especially considering the cross-section of an actual device implemented in all six available metal layers (drawn to scale) at the bottom of Fig. 4. The LSD loop couples flux directly into the MSD loop and it can reach the output not only via the inductive ladder, but also via this magnetic connection to the (strongly coupled to the output) MSD. In addition, the MSD flux can reach the output directly, not mediated by the washer (and with the sign opposite to the washer-mediated coupling).

We treat a complete Φ-DAC structure as a three-port device (LSD, MSD, and OUT) and, using the 3D inductance extraction program `FastHenry` with superconductor support [19], extract its complete inductance matrix

$$\begin{pmatrix} L_{\mathrm{LSD}} & M_{\mathrm{LSD,MSD}} & M_{\mathrm{LSD},OUT} \\ M_{\mathrm{LSD,MSD}} & L_{\mathrm{MSD}} & M_{\mathrm{MSD,OUT}} \\ M_{\mathrm{LSD,OUT}} & M_{\mathrm{MSD,OUT}} & L_{\mathrm{OUT}} \end{pmatrix}.$$

For subsequent analysis we treat the SFQ pulse sources as simple current sources that can produce (up to) I_{in} (approximately half of their junction critical current I_c, as discussed below) into a large inductive load, and calculate all relevant parameters of our Φ-DACs as shown in Table I. After we build a Φ-DAC layout model, we iterate over its geometrical parameters to ensure that it fits into the available space, has the required number of bits and range, and that its MSD/LSD division ratio is such that the LSD comfortably spans a single MSD step.

B. Φ-DAC: SFQ Pulse Sources

Our implementation of an SFQ source is based on perhaps the earliest incarnations of single-flux-quanta circuits [20]: a current biased dc-SQUID made with two shunted junctions.

A schematic of two dc-SQUIID SFQ pulse sources feeding the LSD and MSD storage loops of a single Φ-DAC is shown in Fig. 5.

To operate, one first applies PWR current bias (biasing all junctions to about half of their I_c). ADDR is then applied,

TABLE I
Φ-DAC PARAMETER CALCULATION

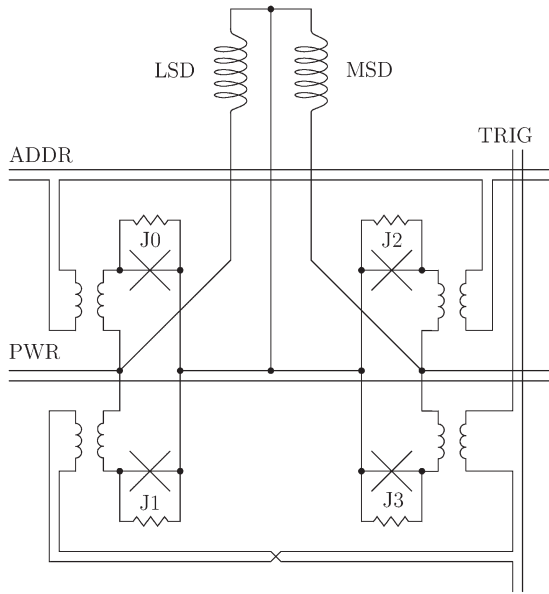|			
Weight of one SFQ in LSD loop, $m\Phi_0/\Phi_0$	W_{LSD}	$=$	$1000 \times \left(\frac{M_{\mathrm{LSD,OUT}}}{L_{\mathrm{LSD}}} - \frac{M_{\mathrm{LSD,MSD}}}{L_{\mathrm{LSD}}} \times \frac{M_{\mathrm{MSD,OUT}}}{L_{\mathrm{MSD}}} \right)$
Weight of one SFQ in MSD loop, $m\Phi_0/\Phi_0$	W_{MSD}	$=$	$1000 \times \frac{M_{\mathrm{MSD,OUT}}}{L_{\mathrm{MSD}}}$
LSD capacity, Φ_0	$\mathrm{MAXSFQ_{LSD}}$	$=$	$\lfloor I_{\mathrm{in}} \times L_{\mathrm{LSD}}/\Phi_0 \rfloor$
MSD capacity, Φ_0	$\mathrm{MAXSFQ_{MSD}}$	$=$	$\lfloor I_{\mathrm{in}} \times L_{\mathrm{MSD}}/\Phi_0 \rfloor$
Division ratio, should be $\leq \mathrm{MAXSFQ_{LSD}}$			$W_{\mathrm{MSD}}/W_{\mathrm{LSD}}$
Total range, $m\Phi_0$	$Range$	$=$	$W_{\mathrm{MSD}} \times \mathrm{MAXSFQ_{MSD}}$
Effective dynamic range, bits			$\log_2 \left| \frac{Range}{W_{\mathrm{LSD}}} \right|$

Fig. 5. Schematic of two SFQ sources feeding two stages of a Φ-DAC. There are 4608 such pairs implemented on a 512-qubit processor. Junction critical current is 55 μA. Each junction is shunted with approximately 0.58, Ω, which corresponds to a $\beta_c \simeq 0.05$. The DAC storage inductances for LSD and MSD loops are $\simeq$1 nH, whereas the inductance of the source itself is 24 pH. PWR, ADDR, and TRIG lines are shared between different Φ-DAC sources as described in Section III-E.

providing an initial flux bias to the dc-SQUID bodies. Ramping TRIG with a polarity that adds to ADDR in, for example, the dc-SQUID comprised of junctions J0 and J1, eventually steers enough current through J0 to exceed its critical current, causing it to "flip" by 2π in phase, admitting a single flux quantum into the dc-SQUID loop. TRIG is then decreased, eventually causing J1 to flip. The J0/J1 dc-SQUID is thus returned to its zero flux state, but in the process the phase drop across the LSD inductor has been increased by 2π—an SFQ pulse is added to that storage loop. Assuming the LSD inductor is large compared to the dc-SQUID inductance, this process can be repeated until the persistent current stored in the LSD loop becomes comparable to the PWR current, cancelling it, preventing junctions from further flipping. At that point, the Φ-DAC loop has reached its maximum SFQ capacity.

If one changes the sign of PWR, using the same process one can add single flux quanta of the *opposite* magnetic field

direction into this storage loop (or, *subtract* from the ones stored there).

Note that the TRIG line is twisted between the dc-SQUIDs J0/J1 and J2/J3, so when it adds to the ADDR pre-bias for the J0/J1 dc-SQUID, it subtracts from the J2/J3 dc-SQUID, and the J2/J3 SQUID is quiescent. But if one reverses the polarity of the TRIG pulses relative to the ADDR pre-bias, one can operate the J2/J3 dc-SQUID, adding a SFQ to the MSD Φ-DAC coil. The relative polarity of ADDR and TRIG allows us to select the Φ-DAC stage on which we want to operate.

The PWR, ADDR, and TRIG levels are chosen to meet the following criteria:

1) With PWR held at its active level, the state of a Φ-DAC changes by exactly one flux quanta per ADDR, TRIG pulse.
2) Each Φ-DAC undergoes SFQ transitions only when all three lines addressing that device are active. If two or fewer lines are active, the state of the Φ-DAC does not change.

In addition, programming a Φ-DAC to a desired state is always done starting from the zero state, which is achieved with a RESET procedure that will be described in the next section.

If the above criteria are met, then a limited number $(O(\sqrt[3]{N}))$ of control lines can address N Φ-DACs in what we call "XYZ" fashion, discussed further in Section III-E. Here, we discuss the process, which we refer to as margining, by which programming levels are chosen to meet the above criteria.

Φ-DAC state stability is fully determined by Φ_b and I_b, where Φ_b is the sum of the ADDR and TRIG flux biases and I_b is the total current biasing the dc-SQUID SFQ pulse source. I_b includes contributions from PWR and from the current circulating in the main Φ-DAC loop due to its flux state. A critical line in (Φ_b, I_b) space, similar to that of a current biased dc-SQUID, bounds the region in which a flux state is stable. When this line is crossed due to manipulation of PWR, ADDR, and TRIG, a transition will take place.

In Fig. 6, the critical line of the zero flux state of the dc-SQUID pulse source is plotted. Crossing this boundary corresponds to the first junction flip in the SFQ pulse sequence described previously. A Φ-DAC will cross the boundary at a point that depends not only on its externally applied biases, but also on the main Φ-DAC loop flux state, which shifts the location of the Φ-DAC in the I_b direction. At any given time, each
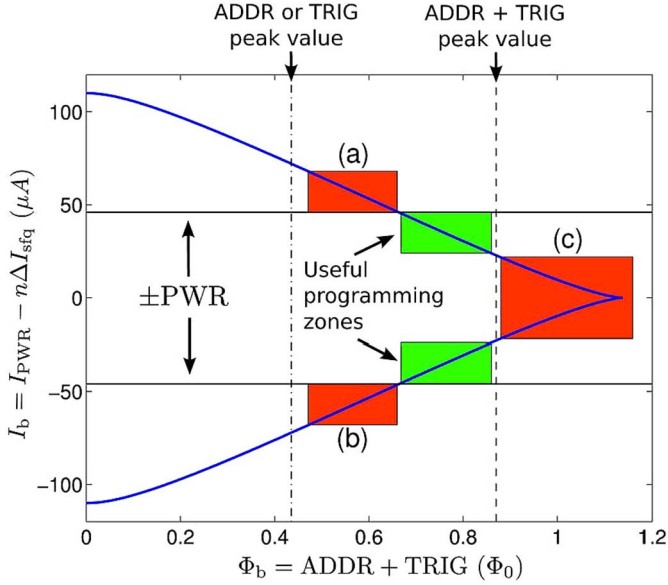
Fig. 6. Φ-DAC margining diagram. The blue curve is the critical line of the dc-SQUID pulse source zero flux state. The active PWR levels for programming $+$SFQ and $-$SFQ are the ± 45 μA horizontal black lines. The red zones must be avoided to respect the margining criteria. Useful programming takes place in the green zones.

Fig. 7.   CAD layout of a single 60×60 μm^2 plaquette of our second generation processor with three Φ-DACs. Note that areas of (top) Φ-DAC storage coils and (bottom) pulse sources (junctions are yellow rectangles) are approximately equal.

Φ-DAC in a processor is positioned on this diagram according to its flux state and which of its bias lines are active. Margining of the PWR, ADDR, and TRIG levels can be understood as a geometric partitioning of the (Φ_b, I_b) space into active regions (green), in which intended transitions will take place, and forbidden zones (red), in which transitions that do not meet the margining criteria would occur.

Φ-DACs addressed by only PWR and either ADDR or TRIG can have I_b values in regions (a) and (b), and therefore ADDR or TRIG levels must not reach these regions. Φ-DACs addressed by only ADDR and TRIG will have I_b values in region (c), and therefore the sum of the ADDR and TRIG levels must not reach this region. The height of region (c) is equal to the combined heights of the two green regions, is equal to the combined heights of the outer red regions, and is equal to the main Φ-DAC loop current range I_{in}. PWR, ADDR, and TRIG levels are chosen to maximize the size of the active regions while avoiding the forbidden zones.

The critical line in Fig. 6 was computed using the average Φ-DAC parameters measured on a D-Wave Two processor. The ADDR and TRIG levels chosen for this processor (vertical dashed lines) do not quite match the boundaries of the allowed and forbidden zones (as would be optimal) due to variation of physical parameters between Φ-DACs and the requirement that the margining criteria be met for all Φ-DACs.

C. Φ-DAC Reset

The protocols described above allow us to add or subtract SFQ to a Φ-DAC stage, ending up with a known number of flux quanta when we start from a known state. For realistic operation we must also be able to reliably reset all Φ-DACs into a known state starting from an *unknown* state.
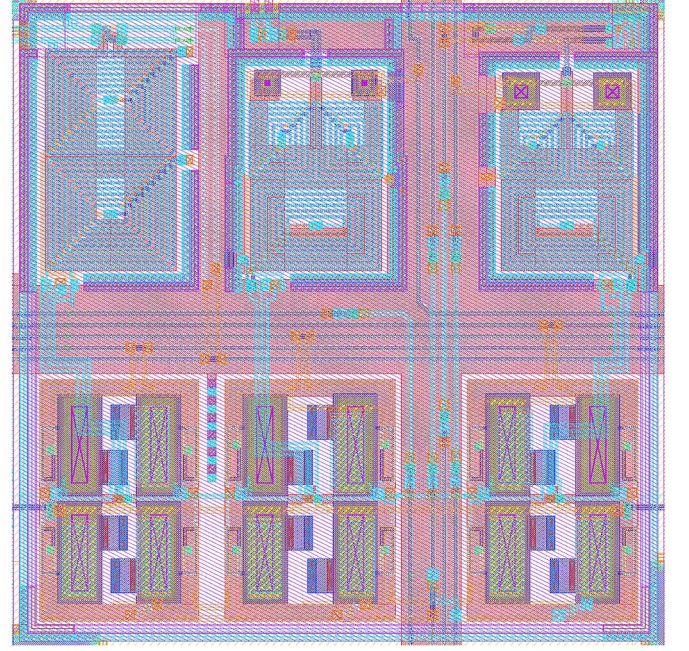
To reset a Φ-DAC, I_{PWR} is set to zero. The dc-SQUID pulse source still sees a non-zero current bias, as long as the Φ-DAC is in a non-zero flux state. Programming the Φ-DAC under these conditions will cause SFQ changes in the main Φ-DAC loop that decrease this current bias. Applying ADDR+TRIG pulses with large enough amplitude to reliably drive transitions (larger than the maximum Φ_b value of the critical boundary in Fig. 6) will 'de-program' the Φ-DAC one SFQ at a time, until it reaches its lowest energy zero SFQ state for which the circulating current is zero. To reliably reach this zero SFQ state, junction critical current asymmetry must be small: the critical current difference between the two junctions in the dc-SQUID pulse source should be well under Φ_0/L, where L is the main loop inductance.

Note that the margining criteria are violated during reset. All Φ-DACs are reset simultaneously.

D. Minimizing Φ-DAC Footprint

As we mentioned in Section II, the Φ-DAC area is what ultimately sets the size of our processor unit tile. This in turn determined the length of the qubits, and thus their energy scales, ultimately affecting the performance of the annealing algorithm. Minimizing this area is therefore of great importance to us.

What matters for a given Φ-DAC to achieve its design objectives is the maximum number of single flux quanta that we can store in its MSD and LSD loops (determining maximum range and precision, provided that division ratio is chosen correctly). That, in turn, is proportional to the LI_c product of the storage loop inductance L and the pulse source junction I_c. How can we minimize an area required to implement both junctions
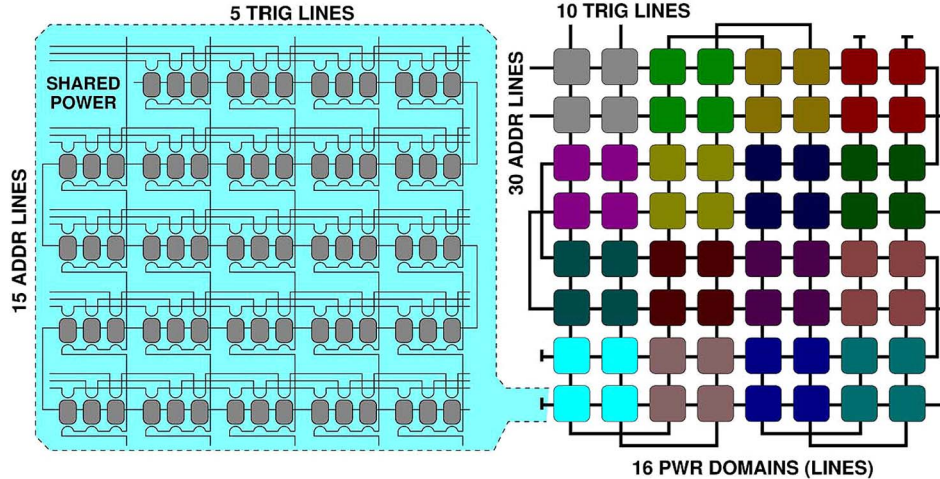
Fig. 8.   XYZ addressing of Φ-DACs in the D-Wave Two processor. (Left) Seventy-two Φ-DACs within a unit tile are selected using 15 ADDR and 5 TRIG lines. (Right) An 8×8 array of unit tiles is split into 16 PWR domains, and all Φ-DACs within that can be addressed using 30 ADDR, 10 TRIG, and 16 PWR lines.

and inductor to achieve constant (and sufficient) LI_c product? Equivalently, how can we maximize this product in given area?

One can observe that (to the first order of approximation), inductance of a spiral coil (for fixed number of available layers) is proportional to its area with some proportionality constant α. The same is true for junction area, given a fixed critical current density J_c. If we have unit area available for the whole Φ-DAC, and inductance occupies some fraction x of that, $LI_c = \alpha x J_c (1-x) \sim x(1-x)$, which reaches its maximum at[3] $x = 0.5$, meaning that half of the optimal Φ-DAC area is occupied by storage inductance and the other half by source junctions.

This was the rule that we used for choosing source junction I_c vs. storage L (their J_c was fixed by requirements of the analog qubit and coupler circuitry in a process with only a single available trilayer). Fig. 7 is a CAD view of three Φ-DACs within one plaquette of our current processor.

Note that the result of this analysis is independent of critical current density. Suppose a second high-J_c trilayer becomes available for our next generation design, say, 9 kA/cm^2 (in addition to our current 250 A/cm^2), a factor of 36 in J_c. Just replacing the existing junctions with smaller in size and equal critical current would save us less than a factor of 2 in Φ-DAC area. If instead L is decreased and I_c is increased by a factor of 6 in value, the total area decreases by the same factor, with LI_c product unchanged.

E. XYZ-Addressing Line Count

We need 72 Φ-DACs to control all the qubits and couplers of a unit tile (6 per qubit, 16 for controlling internal couplers, and 8 for controlling external couplers), for a total of 4608 Φ-DACs for our D-Wave Two 512-qubit processors. To select one of them using cubic XYZ-addressing, we need at least $\lceil 3\sqrt[3]{4608} \rceil = 50$ lines, or about 16 lines per dimension.

---

[3]$(d/dx)x(1-x) = 1-2x = 0 \Rightarrow x = 0.5$; $(d^2/dx^2)x(1-x) = -2 < 0$, thus the maximum is at $x = 0.5$.
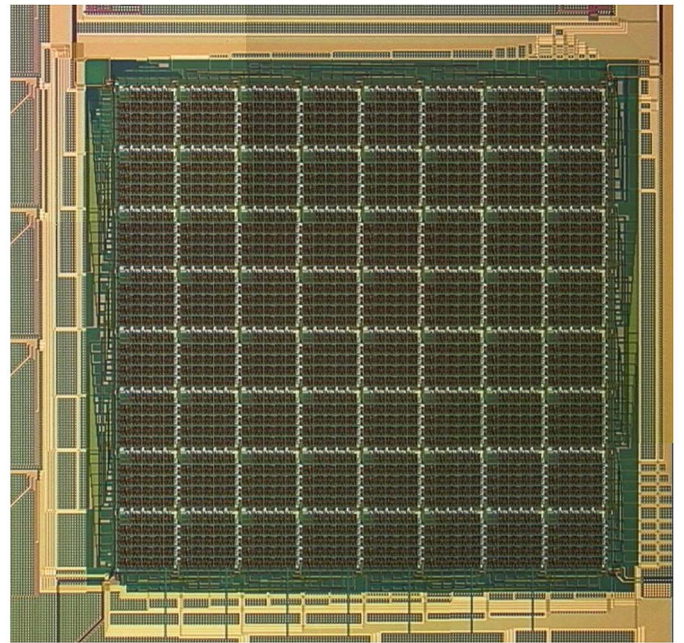

Fig. 9.   Microphotograph of an active portion ($\sim 3.5 \times 3.5$ mm^2) of a D-Wave Two processor chip, 8×8 array of 8-qubit unit tiles; one unit tile is 335 μm on the side. This picture was taken before deposition of the last metal layer (serving as skyplane), making internal structure visible.

We have arranged all required Φ-DACs for a given tile in 25 3-DAC plaquettes (one plaquette is empty), as shown in the left panel of Fig. 8. One of three Φ-DACs within a plaquette is selected using one of three ADDR lines, with all three sharing a TRIG line, resulting in 15 ADDR and 5 TRIG lines addressing all Φ-DACs within unit tile.

The third dimension of addressing is established by separating tile arrays into PWR domains. Our D-Wave Two processors contain an 8×8 array of unit tiles, split into sixteen 2×2 power domains, as shown in the right panel of Fig. 8. All Φ-DACs within one power domain are connected in series and fed by one of 16 PWR lines. 30 ADDR and 10 TRIG lines are reused between power domains, for a total of $30 + 10 + 16 = 56$ lines used to address all Φ-DACs within a processor

matrix. While it is not optimal (because of the difference in the number of ADDR and TRIG lines), it is sufficiently close, and this arrangement allowed us to achieve a more regular layout without having to assign different roles to a single line within the processor fabric (e.g., make a single line work as an ADDR for one DAC and a TRIG for another).

IV. CONCLUSION

We have described how, starting with top-level requirements of a processor implementing a quantum annealing algorithm, we have designed its hardware graph and required control infrastructure, which allowed us to successfully operate processors with up to 512 rf-SQUID qubits using only 56 control lines for problem programming. Fig. 9 shows a microphotograph of an active area of a D-Wave Two processor chip.

The most important feature of our new Φ-DAC design is its zero static power dissipation—unlike traditional SFQ circuitry, which incorporates on-chip resistive current sources tapping a common voltage rail, this design biases all devices serially with a fixed current whose magnitude is set by a *room-temperature* resistor.[4] The only energy dissipated on-chip is on the order of $I_c \times \Phi_0$ per flux quantum moved into (or out of) the storage inductor. For a pair of 55 μA Φ-DAC junctions this corresponds to 0.22 aJ. Complete reprogramming of all 9216 Φ-DAC stages moving from -16 to $+16$ SFQ in their storage loops would dissipate on chip only about 65 fJ.

While the D-Wave One required a post-programming delay of about 1 s, D-Wave Two can thermalize to 20 mK within 10 ms, a factor of 100 improvement achieved within one processor generation just in this post-programming thermalization time.

REFERENCES

[1] R. Harris *et al.*, "Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor," *Phys. Rev. B, Condens. Matter*, vol. 82, no. 2, pp. 024511-1–024511-15, Jul. 2010. [Online]. http://link.aps.org/doi/10.1103/PhysRevB.82.024511

[2] N. Jones, "Google and NASA snap up quantum computer," *Nature News*, May 16, 2013. [Online]. Available: http://stacks.iop.org/0953-2048/23/i=6/a=065004

[3] M. W. Johnson *et al.*, "A scalable control system for a superconducting adiabatic quantum optimization processor," *Supercond. Sci. Technol.*, vol. 23, no. 6, p. 065004, Jun. 2010. [Online]. Available: http://stacks.iop.org/0953-2048/23/i=6/a=065004

[4] A. J. Berkley *et al.*, "A scalable readout system for a superconducting adiabatic quantum optimization system," *Supercond. Sci. Technol.*, vol. 23, no. 10, p. 105 014, Oct. 2010. [Online]. Available: http://stacks.iop.org/0953-2048/23/i=10/a=105014

[5] M. W. Johnson *et al.*, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, May 2011.

[6] S. Boixo *et al.*, "Quantum annealing with more than one hundred qubits," *Nature Phys.*, vol. 10, pp. 1–23, 2013, arXiv:1304.4595.

[7] K. L. Pudenz, T. Albash, and D. A. Lidar, "Error corrected quantum annealing with hundreds of qubits," *Nature Comm.*, vol. 5, pp. 1–18, 2013, arXiv:1307.8190v1.

[8] Z. Bian, F. Chudak, W. G. Macready, L. Clark, and F. Gaitan, "Experimental determination of Ramsey numbers," *Phys. Rev. Lett.*, vol. 111, no. 13, pp. 130505-1–130505-6, Sep. 2013. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.111.130505

[9] H. Neven *et al.*, "NIPS 2009 demonstration: Binary classification using hardware implementation of quantum annealing," in *Proc. NIPS, Demo. (Quantum)*, 2009, pp. 1–17.

[10] R. Harris *et al.*, "Experimental demonstration of a robust and scalable flux qubit," *Phys. Rev. B, Condens. Matter*, vol. 81, no. 13, pp. 134510-1–134510-19, Apr. 2010. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevB.81.134510

[11] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson junction technology for sub-teraherz clock frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.

[12] P. Bunyk, K. Likharev, and D. Zinoviev, "RSFQ technology: Physics and devices," *Int. J. High Speed Electron. Syst.*, vol. 11, no. 1, pp. 257–305, Mar. 2001.

[13] F. Barahona, "On the computational complexity of Ising spin glass models," *J. Phys. A, Math. Gen.*, vol. 15, no. 10, p. 3241, Oct. 1982. [Online]. Available: http://stacks.iop.org/0305-4470/15/i=10/a=028

[14] S. Istrail, "Statistical mechanics, three-dimensionality and np-completeness: I. Universality of intracatability for the partition function of the Ising model across non-planar surfaces," in *Proc. 22nd Annu. ACM Symp. Theory Comput.*, 2000, pp. 87–96.

[15] V. Choi, "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem," *Quantum Inf. Process.*, vol. 7, no. 5, pp. 193–209, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1007/s11128-008-0082-9

[16] A. Maassen van den Brink, A. J. Berkley, and M. Yalowsky, "Mediated tunable coupling of flux qubits," *New J. Phys.*, vol. 7, no. 1, p. 230, Nov. 2005. [Online]. Available: http://stacks.iop.org/1367-2630/7/i=1/a=230

[17] R. Harris *et al.*, "Sign- and magnitude-tunable coupler for superconducting flux qubits," *Phys. Rev. Lett.*, vol. 98, no. 17, pp. 177001-1–177001-4, Apr. 2007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.98.177001

[18] R. Harris *et al.*, "Compound Josephson-junction coupler for flux qubits with minimal crosstalk," *Phys. Rev. B, Condens. Matter*, vol. 80, no. 5, pp. 052506-1–052506-4, Aug. 2009.

[19] S. R. Whiteley. FastHenry 3.0wr. [Online]. Available: http://www.wrcad.com/ftp/pub/README.FASTHENRY

[20] J. P. Hurrell, D. C. Pridmore-Brown, and A. H. Silver, "Analog-to-digital conversion with unlatched squid's," *IEEE Trans. Electron Devices*, vol. ED-27, no. 10, pp. 1887–1896, Oct. 1980.

P. I. Bunyk (M'14) was born in Moscow, Russia. He received the M.Sc. degree (*summa cum laude*) in physics from Moscow State University, Moscow, in 1993 and the M.Sc. degree in computer science from the State University of New York at Stony Brook, NY, USA, in 1997.

In 1993, he joined the State University of New York at Stony Brook to work on application of high-speed single-flux quantum (SFQ) digital logic to high-performance computing under the supervision of Prof. K. K. Likharev. In 2000, he joined the Superconductor Electronics Organization, TRW, Inc. (now Northrop Grumman Corporation), Manhattan Beach, CA, USA, where he continued his work on SFQ microprocessor prototype FLUX-1, as well as superconductor digital signal processing circuitry and high-speed SFQ network switch designs. Subsequently, he worked on design and establishing computer-aided design flows for high-speed analog and digital devices based on semiconductor InP transistor technology. In 2005, he joined the Processor Development Team, D-Wave Systems, Inc., Burnaby, BC, Canada, and has since been working with his colleagues there to develop D-Wave's quantum annealing processor.

Emile M. Hoskinson received the B.Sc. degree from the University of British Columbia, Vancouver, BC, Canada, in 1999 and the Ph.D. degree from the University of California, Berkeley (UC Berkeley), CA, USA, in 2005.

After postdoctoral appointments with Institute N, Grenoble, France, and at UC Berkeley, he joined the Processor Development Team, D-Wave Systems, Inc., Burnaby, BC, in 2011.

Mark W. Johnson (M'12) was born in Santa Barbara, CA, USA. He received the B.S. degree (with honors) in physics from Harvey Mudd College, Claremont, CA, in 1989 and the M.S. and Ph.D. degrees in physics from the University of Rochester, Rochester, NY, USA, in 1991 and 1996, respectively.

He worked for several years in the Superconductor Electronics Organization, TRW, Inc. (now Northrop Grumman Corporation), Manhattan Beach, CA, developing superconductor analog-to-digital converters and digital signal processing circuitry in Northrop Grumman's Nb- and NbN-based integrated circuit technologies. In 2005, he joined the Processor Development Team, D-Wave Systems, Inc., Burnaby, BC, Canada, and has since been working with his colleagues there to develop D-Wave's quantum annealing processor.

---

[4]This approach can also be viewed as implementing SFQ "current recycling" taken to its ultimate limit.

Elena Tolkacheva received the M.Sc. degree in physics from Moscow State University, Moscow, Russia, in 2002 and the Licentiate degree in engineering from Chalmers University of Technology, Gotenborg, Sweden, in 2005, for her work on implementing digital signal processor parts for wireless communications using single-flux quantum logic.

Subsequently, she was a Postdoctoral Researcher with Physikalisch-Technische Bundesanstalt, investigating SFQ and Josephson electronics for metrological applications and phase and flux qubit manipulation and control circuitry. In 2007, she joined the Design and Layout Team within the Processor Development Team, D-Wave Systems, Inc., Burnaby, BC, Canada.

Fabio Altomare received the Laurea degree in physics from the University of Pisa, Pisa, Italy, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA.

After working as a Postdoctoral Research Associate at Duke University, Durham, NC, USA, and at the National Institute of Standards and Technology, he joined D-Wave Systems, Inc., Burnaby, BC, Canada, where he is involved in the practical implementation of a quantum annealing processor.

Andrew J. Berkley received the B.Sc. degree in applied and engineering physics from Cornell University, Ithaca, NY, USA, in 1997 and the Ph.D. degree in physics from the University of Maryland, College Park, MD, USA, in 2003.

Since 2004, he has been with D-Wave Systems, Inc., Burnaby, BC, Canada, as an Experimental Physicist.

Richard Harris received the B.Sc. degree in physics from McMaster University, Hamilton, ON, Canada, in 1997 and the Ph.D. degree in physics from the University of British Columbia, Vancouver, BC, Canada, in 2003.

He was a Natural Sciences and Engineering Research Council (Canada) Postdoctoral Fellow with Stanford University, Stanford, CA, USA, from 2004 to 2006 prior to joining D-Wave Systems, Inc., Burnaby, BC, as an Experimental Physicist in 2006.

Jeremy P. Hilton was born in Kitchener, ON, Canada. He received the B.Sc. degree in physics from the University of British Columbia, Vancouver, BC, Canada, in 1999.

In 2000, he joined D-Wave Systems, Inc., Burnaby, BC, where he has focused on technology development in the context of superconducting quantum computing. During his tenure at D-Wave, he has provided leadership in the development of proof-of-concept quantum annealing processors, production-level superconductor integrated circuit fabrication, ultralow-temperature and magnetic vacuum operating environments, and integration of classical and quantum superconductor ICs for the realization of scalable quantum computing, including 128- and 512-qubit quantum annealing processors.

Trevor Lanting received the B.Sc. degree in physics and astronomy from the University of British Columbia, Vancouver, BC, Canada, in 2000 and the Ph.D. degree in physics from the University of California, Berkeley, CA, USA, in 2006.

He has been an Experimental Physicist with D-Wave Systems, Inc., Burnaby, BC, since 2008.

Anthony J. Przybysz received the Ph.D. degree in physics from the University of Maryland, College Park, MD, USA, in 2010, where he worked on reducing decoherence in dc superconducting quantum interference device phase qubits under Dr. F. Wellstood.

In 2010, he joined with D-Wave Systems, Inc., Burnaby, BC, Canada, calibrating and benchmarking the performance of the D-Wave Two adiabatic quantum optimization processor and demonstrating equilibrium entanglement of up to eight flux qubits. He has been with Northrop Grumman Corporation, Manhattan Beach, CA, USA, since March 2013, designing and characterizing transmon, flux, and phase qubits.

Jed Whittaker received the B.S. and M.S. degrees in physics from Brigham Young University, Provo, UT, USA, in 2003 and 2004, respectively, and the Ph.D. degree in physics from the University of Colorado at Boulder, CO, USA, in 2012, where he studied superconducting qubit physics.

He has been an Experimental Physicist with D-Wave Systems, Inc., Burnaby, BC, Canada, since 2013.