A Cost and Power Feasibility Analysis of Quantum Annealing for NextG Cellular Wireless Networks

SRIKAR KASI1,2, P. A. WARBURTON3, JOHN KA EWELL2, KYLE JAMIESON1

1Princeton University, NJ 08542, USA
2InterDigital, Inc., PA 19428, USA
3University College London, WC1E 6BT, UK

Corresponding author: Srikar Kasi (email: skasi@princeton.edu).

ABSTRACT In order to meet mobile cellular users’ ever-increasing data demands, today’s 4G and 5G wireless networks are designed mainly with the goal of maximizing spectral efficiency. While they have made progress in this regard, controlling the carbon footprint and operational costs of such networks remains a long-standing problem among network designers. This paper takes a long view on this problem, envisioning a NextG scenario where the network leverages quantum annealing for cellular baseband processing. We gather and synthesize insights on power consumption, computational throughput and latency, spectral efficiency, operational cost, and feasibility timelines surrounding quantum annealing technology. Armed with these data, we project the quantitative performance targets future quantum annealing hardware must meet in order to provide a computational and power advantage over CMOS hardware, while matching its whole-network spectral efficiency. Our quantitative analysis predicts that with 82.32 µs problem latency and 2.68M qubits, quantum annealing will achieve a spectral efficiency equal to CMOS while reducing power consumption by 41 kW (45% lower) in a Large MIMO base station with 400 MHz bandwidth and 64 antennas, and an 160 kW power reduction (55% lower) using 8.04M qubits in a CRAN setting with three Large MIMO base stations.

INDEX TERMS Quantum annealing, quantum computing, radio access networks, wireless communication

I. INTRODUCTION Radio Access Networks (RANs) are experiencing unprecedented growth in traffic at base stations due to increased subscriber numbers and their higher quality of service requirements [1]. To meet the resulting demand, 5G and NextG RANs are expected to deploy sophisticated techniques such as cell densification, multiple-input multiple-output communication, and millimeter-wave communication [2]. But this significantly increases the power and cost required to operate RANs backed by complementary metal oxide semiconductor (CMOS)-based processing. While general energy-saving strategies such as sleep mode [3] and network planning [4] can be used to decrease RAN’s power consumption to a point, the fundamental problem of power requirements scaling with the exponentially increasing computational requirements of the RAN persists. Previously (ca. 2010), this problem had not limited innovation in the design of RANs, due to a rapid pace of improvement in CMOS’s computational efficiency—which has typically followed Dennard scaling [5]–[7] for power consumption. Unfortunately however, today, such improvements are becoming increasingly difficult to maintain, due to transistor sizes approaching atomic limits, and issues such as leakage current control and thermal runaway [8]. As a result, CMOS operational clock speeds have reached a plateau and Moore’s Law scaling has come to an end (ca. 2025–2030) [9]–[11]. This therefore calls into question the prospects of CMOS to handle NextG cellular demand in terms of both energy and spectral efficiency. While unanticipated advances in CMOS may allow it to handle this demand, this paper makes the case for the possible future feasibility and potential power advantage of quantum annealing, a candidate quantum technology, over CMOS, in certain RAN operation scenarios.

Recently quantum computers previously only hypothesized have been commercialized [12]–[14], and are now available for use by researchers. The current and near-term quantum technology can be broadly classified into digital gate-model and analog annealing-model architectures [15]–[18]. Gate-model devices are fully general purpose computers, using programmable logic gates acting on qubits, whereas annealing-model devices are specialized computers, offering a means to...
search an optimization problem for its lowest energy configurations in a high-dimensional energy landscape [18]. While gate-model devices of size relevant to practical applications are not yet generally available [19], today’s annealing-model devices with about 5,000 qubits enable us to commence empirical studies at realistic scales [16]. In particular there are several published proof-of-principle studies of using quantum annealing to solve computational problems in communications networks [20]–[31]. Therefore we conduct this study from the perspective of annealing-model devices.

Here we present the first extensive analysis on power consumption and quantum annealing (QA) architecture to make the case for the future feasibility of quantum processing based RANs. We seek to quantitatively analyze whether in the coming years and decades, mobile operators might rationally invest in the RAN’s capital expenditure (CapEx) by purchasing quantum hardware of high cost, in a bid to lower its operational expenditure (OpEx) and hence the Total Cost of Ownership (CapEx + OpEx). The OpEx cost reduction would result from the reduced power consumption of the RAN, due to higher computational efficiency of quantum processing over CMOS processing for certain heavyweight baseband computational tasks. Unlike CMOS devices, the power consumption of quantum devices is dominated by their refrigeration unit rather than the computation at hand [32]–[34], implying that the increasing computational demand in RANs will have negligible impact on power consumption. Note that nothing which we propose with quantum annealing here is fundamentally out of the reach of classical computation. The potential advantages of QA for RAN applications are purely economic (i.e., the lower cost of operation resulting from the lower power consumption). Figure 1 depicts our envisioned scenario, where quantum processing units (QPUs) co-exist with CMOS processing units (CPUs) at Centralized RAN (CRAN) Baseband Units (BBUs). QPUs will then be used for the BBU’s heavy baseband processing, whereas CPUs will handle the network’s lightweight processing such as the control plane (e.g., resource allocation), and pre-/post-processing the QPU-specific computation.

While recent successful point-solutions that apply QA to a variety of wireless applications [20]–[31] serve as our motivation, previous work stops short of a holistic power and cost comparison between QA and CMOS. Despite QA’s benefits demonstrated by these prior works in their respective point settings, a reasoning of how these results will factor into the overall computational performance and power requirements of the base station and CRAN remains lacking. Therefore, here we investigate these issues head-on, to make an end-to-end case that QA will likely offer benefits over CMOS for handling BBU processing, and to make time predictions on when these benefits might be realized. Specifically, we present informed answers to the following questions:

**Question 1:** How many qubits are required to realize a base station or CRAN BBU processing requirements? (Answer: cf. §V, §VII)

**Question 2:** Given sufficient number of qubits, how much power and cost does QA save over CMOS? (Answer: cf. §VI)

**Question 3:** At what year might these qubit numbers become feasible, based on the current industry trends? (Answer: cf. §VII)

**Question 4:** How does QA processing latency and solution accuracy impact the qubit requirement and power/cost benefits? (Answer: cf. §III, §V, §VI)

**Question 5:** In what wireless network scenarios QA will provide power/cost advantage over CMOS? (Answer: cf. §VI, VII)

In order to answer the above questions, several key performance indicators need to be analyzed, quantified, and evaluated, most notably the computational throughput and latency (§III), the power consumption of the entire system and resulting spectral efficiency (bits per second per Hertz of frequency spectrum) and operational cost (§VI). We first describe the factors that influence processing latency and throughput on current QA devices and then, by assessing recent developments in the area, project what computational throughput and latency future QA devices can achieve (§III). We analyze cost by evaluating the power consumption of QA and CMOS-based processing at equal spectral efficiency targets (§VI). Our analysis reveals that a three-way interplay between latency, power consumption, and qubit count available in the QA hardware determines whether QA can benefit over CMOS. In particular, latency influences spectral efficiency, power consumption influences energy efficiency, and the number of qubits influences both. Based on these insights, we determine properties that QA hardware must meet in order to provide an advantage over CMOS in terms of energy, cost, and spectral efficiency in wireless networks.

Table 1 summarizes our results, showing that for 200 and 400 MHz bandwidths, respectively, with 1.34M and 2.68M qubits, we predict that QA processing will achieve spectral efficiency equal to today’s 14 nm CMOS processing, while reducing power consumption by 8 kW (16% lower) and
Table 1: QA qubit count requirements to achieve equal spectral efficiency to CMOS, and power consumption of CMOS and QA.\(^1\) Shaded cells indicate the lesser power of CMOS vs. QA.

<table>
<thead>
<tr>
<th>B/W</th>
<th>Qubits</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BS</td>
<td>CRAN BS (kW)</td>
</tr>
<tr>
<td></td>
<td>CMOS</td>
<td>QA</td>
</tr>
<tr>
<td></td>
<td>CRAN</td>
<td>QA</td>
</tr>
<tr>
<td>50 MHz</td>
<td>335K</td>
<td>1.00M</td>
</tr>
<tr>
<td>100</td>
<td>669K</td>
<td>2.00M</td>
</tr>
<tr>
<td>200</td>
<td>1.34M</td>
<td>4.02M</td>
</tr>
<tr>
<td>400</td>
<td>2.68M</td>
<td>8.04M</td>
</tr>
</tbody>
</table>

41 kW (45% lower) in representative 5G/NextG base station scenarios. In a CRAN setting with three base stations of 200 and 400 MHz bandwidths, QA processing with 4.02M and 8.04M qubits, respectively, reduces power consumption by 70 kW (41% lower) and 160 kW (55% lower), while achieving equal spectral efficiency to CMOS.

Our further evaluations compare QA against future 1.5 nm CMOS, which is expected to be the silicon technology at the end of Moore’s Law scaling [9]–[11]. In a CRAN setting with three 400 MHz bandwidth 64-antenna base stations, QA with 8M qubits will reduce power consumption by 23.6 kW (21% lower) while achieving equal spectral efficiency to CMOS.

A projected QA feasibility timeline is reported in Figure 14, describing year-by-year milestones on the application of QA for wireless networks (see §VII). Our analysis shows that with QA qubit connectivity matching the problem connectivity (see §V) and qubits growing 2.65× every three years (the 2017–2020 trend), a power/cost benefit of QA over CMOS is a predicted 11–14 years (ca. 2034–2037) away, whereas the feasibility in processing for a small base station with 10 MHz bandwidth and 32 antennas is a predicted three years away.

Overall, our quantitative results show that QA will offer power/cost benefits over CMOS in certain wireless network scenarios, once QA hardware scales to at least 537K qubits (§VII) while reducing problem processing time to tens of microseconds, which we argue is feasible within our projected timelines. Scaling of QA processors hold challenges related to engineering, control, and operation of hardware resources, which designers continue to investigate [35], [36]. Recent work demonstrates large-scale qubit control techniques, showing that control of million qubit-scale quantum hardware is already at this point in time a realistic prospect [37].

II. BACKGROUND

In this section, we provide background on 5G/NextG wireless architecture (§II-A) and Quantum Annealing (§II-B).

A. MASSIVE/LARGE MIMO NEXTG ARCHITECTURE

Today’s wireless industry is facing significant challenges in handling mobile cellular traffic at base stations (BSs) due to sharp rises in user counts and their network usage. To meet the resulting demand, the baseband unit (BBU) processing (i.e., digital processing) from many BSs is being aggregated into centralized locations, a concept referred to as a Centralized Radio Access Network or CRAN [38], [39]. This has two immediate advantages: first, compute resources previously dedicated to each BS can be statistically multiplexed among many BSs, saving energy and reducing cost, and second, joint computational processing over the signals to or from many BSs is simplified, since each BS’s processing occurs on either exactly the same physical servers, or physical servers in close network proximity. Despite these advantages, however, CRAN BBUs need to process heavy computational loads within a threshold turnaround time, imposing additional latency and bandwidth requirements on the interconnect between BSs and the centralized BBU.

In 5G and NextG CRAN networks, BSs are envisioned with Multiple-Input Multiple-Output (MIMO) communication, a spatial multiplexing technique typically implemented using multiple antennas at the BS. MIMO communication is a key requirement to enable high spectral efficiency networks envisioned in 5G and NextG [40]–[42]. The status quo implementation, called Massive MIMO, uses a number of antennas (typically 4 or 8 in 5G) for capturing the same user signal, and so to support more users simultaneously, Massive MIMO demands significantly more antennas at the BS [41]. To address this problem, NextG Large MIMO techniques are underway, which use one antenna for the same task, increasing the number of simultaneous users, thus maximizing the wireless network’s spectral efficiency [43].

Typical real-world BS and CRAN implementations involves performance sacrifices which arise due to the strict timing deadline (0.5–1 ms in 5G) by which wireless signals must turnaround. Most notably, this includes the use of linear/low-complexity algorithms, reduced bit precision, and limiting the count of iterative procedures, which all sacrifice spectral efficiency. While Maximum-Likelihood (ML) methods are known to provide optimal performance by maximizing spectral efficiency, they are of exponential computational complexity and so challenging to realize on CMOS hardware. Recent prior work in this area has shown QA to be a promising alternative to CMOS in this regard, realizing ML methods on the order of hundreds of microseconds (excluding overheads) [20], [22], [23]. In our evaluations, we compare the cost/power of QA and CMOS in both Massive and Large MIMO BS and CRAN networks with non-linear MIMO settings (see §VI).

B. QUANTUM ANNEALING

Quantum Annealing is an optimization-based approach that aims to find the lowest energy spin configuration (i.e., solution) of an Ising model described by the time-dependent energy functional (Hamiltonian):

$$H(s) = -\Gamma(s)H_I + L(s)H_P \quad (1)$$

where $H_I$ is the initial Hamiltonian, $H_P$ is the (input) problem Hamiltonian, $s \in [0, 1]$ is a non-decreasing function of time.
called an annealing schedule, $\Gamma(s)$ and $L(s)$ are energy scaling functions of the transverse and longitudinal fields in the annealer respectively. Essentially, $\Gamma(s)$ guides the probability of quantum tunneling during the annealing process, and $L(s)$ guides the probability of finding the ground state of the input problem Hamiltonian $H_P$ [16]. The QA hardware is a network of locally interacting radio-frequency superconducting qubits, organized in groups of unit cells. Fig. 2 shows the unit cell structures of recent (Chimera) and state-of-the-art (Pegasus) QA devices. The nodes and edges in the figure are qubits and couplers respectively [20].

The process of optimizing a problem in the QA is called annealing. Starting with a high transverse field (i.e., $\Gamma(0) \gg L(0) \approx 0$), QA initializes the qubits in a pre-known ground state of the initial Hamiltonian $H_I$, then gradually interpolates this Hamiltonian over time—decreasing $\Gamma(s)$ and increasing $L(s)$—by adiabatically introducing quantum fluctuations in a low-temperature environment, until the transverse field diminishes (i.e., $L(1) \gg \Gamma(1) \approx 0$). This time-dependent interpolation of the Hamiltonian is essentially the quantum annealing algorithm. The Adiabatic Theorem then ensures that by interpolating the Hamiltonian slowly enough, the system remains in the ground state of the interpolating Hamiltonian [45]. Thus during the annealing process, the system ideally stays in a local minimum and probabilistically reaches the global minimum of the problem Hamiltonian $H_P$ at its conclusion [16]. The initial and problem Hamiltonians take the form $H_I = \sum_i \sigma_i^x$ and $H_P = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z$, where $\sigma_i^{x,z}$ are the Pauli spin operators acting on the $i^{th}$ qubit, $h_i$ and $J_{ij}$ are the optimization problem inputs (coefficients) that the user supplies [16].

**Input Problem Forms.** QAs optimize Ising model problems, whose problem format matches the above problem Hamiltonian: $E = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$, where $E$ is the energy of the candidate solution, $s_i$ is the $i^{th}$ solution variable which can take on values in $\{-1, 1\}$, $h_i$ and $J_{ij}$ are called the bias of $s_i$ and the coupling strength between $s_i$ and $s_j$, respectively. Ising form is equivalent to quadratic unconstrained binary optimization (QUBO) form, where solution variables take values in $\{0, 1\}$. Biases represent individual preferences of qubits to take on a particular classical value ($-1$ or $+1$), whereas coupling strengths represent pairwise preferences (i.e., two particular qubits should take on same/opposite values), in the solution the machine outputs. Biases and coupling strengths are specified to qubits and couplers, respectively, using a programmable on-chip control circuitry [46], [47]. The QA probabilistically returns the solution variable configuration with the minimum energy $E$ at its output [20].

**Assumption 1 — Ising Model formulation.** To enable QA computation, cellular baseband’s heavy processing tasks must be formulated as Ising model problems. Recent prior work in this area has formulated the most heavyweight tasks in the baseband, such as frequency domain detection, forward error correction, and precoding problems, into Ising models [20], [21], [23], [30], [31], [48]. Further baseband tasks will either admit Ising model formulations via binary representation of continuous values [49] (we leave for future work), or are so lightweight they require negligible power.

**C. INPUT PROBLEM EMBEDDING**

The process of mapping a given input problem onto the physical QA hardware is called embedding. To understand embedding, let us consider an example Ising problem:

$$E = h_1 s_1 + h_2 s_2 + h_3 s_3 + J_{12} s_1 s_2 + J_{23} s_2 s_3 + J_{13} s_1 s_3 \tag{2}$$

The logical representation of Eq. 2 is depicted in Fig. 3(a), where nodes and edges are qubits and couplers respectively. The curved arrows are used to visualize the linear coefficients. However, observe that a complete three-node qubit connectivity does not exist in the Chimera graph (cf. Fig. 2(a)). Hence the standard approach is to map one of the logical problem variables (e.g., $q_3$) onto two physical qubits (e.g., $q_{3a}$ and $q_{3b}$) as in Fig. 3(b), such that the resulting connectivity can be realized on the native QA hardware. To ensure proper embedding: $q_{3a}$ and $q_{3b}$ must agree with each other. This is achieved by enforcing the condition $h_3 = h_{3a} + h_{3b}$, and chaining these physical qubits with a strong ferromagnetic coupling strength called $J_{Ferro}$ ($J_F$)—see dotted line in Fig. 3(b). The physical Ising problem the QA optimizes for the example in Eq. 2 is then:

$$E = h_1 q_1 + h_2 q_2 + h_3 a q_{3a} + h_3 b q_{3b} + J_{12} q_1 q_2 + J_{13} q_1 q_{3a} + J_{23} q_2 q_{3b} + J_{F} q_{3a} q_{3b} \tag{3}$$
Since $J_F$ is finite, some parameter optimization may be necessary [50], [51].

**Assumption 2—Bespoke QA hardware.** Qubit connectivity significantly impacts performance, with sparse qubit connectivity negatively affecting dense problem graphs due to problem mapping difficulties [20]. Recent advances in QA have bolstered qubit connectivity—a 6 to 15 to 20 couplers per qubit in the Chimera (2017), Pegasus (2020), and Zephyr (ca. 2023-24) topologies respectively [52], [53]—and further improvement efforts continue [54], [55], which will allow QA hardware tailored to baseband processing problems within the timescales of our predictions, resulting in a highly efficient embedding process (see §V-B for a more detailed discussion).

### III. QUANTUM PROCESSING PERFORMANCE

To characterize current and future QA performance, this section analyzes processing time on QA devices, the client of which sends quantum machine instructions (QMI) that characterize an input problem computation to a QA QPU. The QPU then responds with solution data. Fig. 4 depicts the entire latency a QMI experiences from entering the QPU to the read-out of the solution, which consists of programming (§III-A), sampling (§III-B), and post-processing (§III-C) times.

#### A. PROGRAMMING TIME

As the QMI reaches the QPU, the QPU programs the QMI’s input problem coefficients—biases and coupling strengths (§II): room temperature electronics send raw signals into the QA refrigerator unit to program the on-chip flux digital-to-analog converters (Φ-DACs). The Φ-DACs then apply external magnetic fields and magnetic couplings locally to the qubits and couplers respectively. This process is called a programming cycle, and in current technology it takes 4–40 µs, dictated by the amount of programming data, bandwidth of control lines, and the Φ-DAC addressing scheme [35], [56]. During the programming cycle, the QPU dissipates an amount of heat that increases the effective temperature of the qubits. This is due to the movement of flux quanta\(^3\) in the inductive storage loops of Φ-DACs. Thus, a post-programming thermalization time is required to cool the QPU, ensure proper reset/initialization of qubits, and allow the QPU to maintain a thermal equilibrium with the refrigeration unit ($\approx$20 mK). QA clients can specify thermalization times in the range 0–10 ms with microsecond-level granularity. The default value on D-Wave’s machine is a conservative one millisecond [16].

1) **Programming: Data and Bandwidth**

An $N_Q$ qubit, $N_C$ coupler, and $K$-bit precision QA device will program a worst-case $D_{prog} = K \cdot (N_Q + N_C)$ amount of data. With an aggregate programming control line bandwidth $BW_{prog}$, this requires a worst-case $D_{prog} / BW_{prog}$ of data programming time. If the $N_Q$ qubits are equally distributed into $N_{chips}$ number of independently controlled chips (physically located under the same refrigeration unit), all chips can be programmed in parallel, scaling the data programming time by a factor of $1/N_{chips}$. Figure 5 reports these results, showing achievable data programming times at various control line bandwidths. To maintain today’s 40 µs data programming time in a 10M qubit QA device, required aggregate programming control line bandwidth is 33 GHz when 20 couplers per qubit are programmed (typical for practical wireless applications).

2) **Programming: Energy and Thermalization Time**

The next step is QPU thermalization. QMI coefficients are programmed by using six Φ-DACs per qubit and one Φ-DAC per coupler [36]. Each Φ-DAC consists two inductor storage loops with a pair of Josephson junctions each. The energy

---

\(^3\)QA devices store coefficient information in the form of magnetic flux quanta and it is transferred via single flux quantum (SFQ) voltage pulses [36].
dissipated on chip is on the order of \( I_c \times \Phi_0 \) per single flux quantum (SFQ) moved in an inductor storage loop, where \( I_c \) is the \( \Phi\)-DAC’s junction critical current and \( \Phi_0 \) is the magnetic flux quantum.\(^4\) Therefore, the dissipated on-chip programming energy \( (E_{\text{prog}}) \) is given by:

\[
E_{\text{prog}} = 4 \times (6N_Q + N_C) \times I_c \times \Phi_0 \times N_{\text{SFQ}} \quad (4)
\]

where \( N_Q \) and \( N_C \) are the number of qubits and couplers being programmed, and \( N_{\text{SFQ}} \) is the number of SFQs moving into (or out of) inductor storage loops. The required programming thermalization time \( (T_{\text{therm}}) \) is then given by:

\[
T_{\text{therm}} = E_{\text{prog}} / P_{\text{QPU}} \quad (5)
\]

where \( P_{\text{QPU}} \) is the cooling power available at the 20 mK QPU stage, which is typically 30 \( \mu \)W [57]. The supported bit-precision on QA devices is currently up to five bits (four for value, one for sign), and so for the worst-case reprogramming scenario, this corresponds to 32 SFQs (−16 to +16) moving into (or out of) all \( \Phi\)-DAC inductor storage loops [36]. Figure 6 reports these results, showing that programming a large-scale QA device with 10M qubits and 20 couplers per qubit will dissipate only 42 pJ energy on chip, requiring a thermalization time of 1.4 \( \mu \)s only.

After programming and thermalization, the next step resets/initializes the qubits (cf. §II-B), during which each qubit transitions from a higher energy state to an intended ground state, generating spontaneous photon emissions, heating the QPU. Reed et al. [59] demonstrate the suppression of these emissions using Purcell filters, requiring 80 ns (120 ns) for 99% (99.9%) fidelity. Heretofore, an overall programming time of 41.52 \( \mu \)s (data programming: 40 \( \mu \)s, thermalization: 1.4 \( \mu \)s, reset: 0.12\( \mu \)s) is considered for a large-scale 10M qubit QA device, which is subject to the requirement of 33 GHz aggregate control line bandwidth and Purcell filter integration.

\(^4\)\( \Phi_0 = h/2e \), where \( h \) is Planck’s constant and \( e \) is the electron charge.

B. SAMPLING TIME

The process of executing a QMI on a QA device is called sampling, and the time taken for sampling is called the sampling time. The sampling time is classified into three sub-components: the anneal, readout, and readout delay times. A single QMI consists of multiple samples of an input problem, with each sample annealed and read out once, followed by a readout delay (see Fig. 4). Sampling a QMI begins after the QPU programming process.

1) Anneal

In this time interval, the QPU implements a QA algorithm (§II-B) [16] to solve the input problem, where low-frequency annealing lines control the annealing algorithm’s schedule. The bandwidth of these control lines limits the minimum annealing time, which is 0.5 \( \mu \)s today. Weber et al. [60] propose the use of flexible print cables with a moderate bandwidth (≈100 MHz) and high isolation (≈50 dB) for annealing, which potentially decrease annealing time to tens of nanoseconds. Further experiments have demonstrated that large-scale QA devices can be operated under 40 ns anneal time, enabling coherent quantum annealing regimes [61, 62].

2) Readout

After annealing, the spin configuration of qubits (i.e., the solution) is read out by measuring the qubits’ persistent current \( I_p \). This readout information propagates from the qubits to readout detectors located at the perimeter of the QPU chip via flux bias lines. Each flux bias line is a chain of electrical circuits called Quantum Flux Parametrons (QFPs), which detect and amplify qubits’ \( I_p \) to improve the readout signal-to-noise ratio. These QFP chains act like shift registers, propagating the information from qubits to detectors [63]. In current QA devices with \( N_Q \) qubits, there are \( \sqrt{N_Q/2} \) flux bias lines, with each flux bias line responsible for reading out \( 2N_Q/2 \) qubits. Further, each flux bias line reads out one qubit at a time (i.e., time-division readout), thus a total of \( \sqrt{N_Q/2} \) qubits are read out in parallel. Hence, the readout time depends on the qubits’ physical locations, the bandwidth of flux bias lines, and the signal integration time. For the current status of technology, the readout time is 25–150 \( \mu \)s per sample [16]. Nevertheless, recent research demonstrates promising fast readout techniques, which we describe next.

Chen et al. [64] and Heinsoo et al. [65] describe frequency-multiplex readout schemes that enable simultaneous readout of multiple qubits within a flux bias line. While there is no fundamental limit on the number of qubits read out simultaneously, a physical limit is imposed by the line width of qubits’ readout microresonators and the 4–8 GHz operating band (6 GHz center frequency, 4 GHz bandwidth) of commercial microwave transmission line components used in the readout architecture [63]. Microresonators with quality factor \( Q_r \) can capture line widths up to 6/Q\(_r\), thus enabling up to \( 4 \times Q_r/6 \) qubits to be read out simultaneously. Table 2 reports these results, showing that a \( Q_r \) of \( 10^6 \) will enable up to \( \approx 666 \) K qubit-parallel readout. This analysis assumes that...
After a sample’s anneal-readout process, a readout delay (88 ns) to achieve a 98.25% (99.2%) readout fidelity. Their µ

Table 2: The table shows the number of qubits read out in parallel by time-division (status quo) and frequency-multiplex (projected) readout schemes at various choices of QPU sizes and readout microresonator quality factors \(Q_r\).

<table>
<thead>
<tr>
<th>Qubits</th>
<th>Qubits readout in parallel</th>
<th>(Q_r = 10^3) [63]</th>
<th>(Q_r = 10^9) [66]</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>16</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>2048</td>
<td>32</td>
<td>(\approx 666)</td>
<td>2,048</td>
</tr>
<tr>
<td>5,436</td>
<td>(\approx 52)</td>
<td>(\approx 666)</td>
<td>5,436</td>
</tr>
<tr>
<td>10 M</td>
<td>(\approx 2,200)</td>
<td>(\approx 666)</td>
<td>(\approx 666)K</td>
</tr>
</tbody>
</table>

each microresonator can be fabricated at exactly its design frequency, which is currently not the case. Further developments in understanding the RF properties of microresonators will be needed to achieve this multiplexing performance.

In order to avoid sample-to-sample readout correlation, microresonators reading out the current sample’s qubits must ring down before reading the next sample’s qubits. McClure et al. [67] achieve ring-down times on the order of hundreds of nanoseconds by applying pulse sequences that rapidly extract residual photons exiting the microresonators after readout. Fast ring-down can also be achieved by switching off the QFP (after the readout) coupled to a microresonator, and then switching on a different QFP that couples the microresonator to a lossy line. While QFP on-off switching takes hundreds of nanoseconds [68], [69], it ensures high fidelity readout.

Recent work by Grover et al. [68] shows the application of QFPs as isolators, achieving a readout fidelity of 98.6% (99.6%) in 80 ns (1 µs) only. Work by Walter et al. [70] describes a single-shot readout scheme requiring only 48 ns (88 ns) to achieve a 98.25% (99.2%) readout fidelity. Their designs are also compatible with multiplexed architectures and earlier readout schemes, implying that by design integration readout time reaches on the order of microseconds per sample.

3) Readout delay

After a sample’s anneal-readout process, a readout delay is added (see Fig. 4). In this time interval, qubits are reset for the next sample’s anneal. QA clients can specify times in the range 0–10 ms, and the default value is a conservative one millisecond. Nevertheless, about one microsecond is sufficient for high fidelity qubit reset (§3.1) [59].

C. POSTPROCESSING TIME

This time interval is used for post-processing the solutions returned by QA for improving the solution quality [71]. Multiple samples’ solutions are post-processed at once in parallel with the current QMI’s annealer computation, whereas the final batch of post-processing occurs in parallel with the programming of next QMI. Thus, the post-processing time does not factor into the overall processing time [56].

In summary, the projected programming time is 41.52 µs (data programming: 40 µs, thermalization: 1.4 µs, reset: 0.12 µs), anneal time is 40 ns/sample, readout time is one µs/sample, and readout delay time is one µs/sample. For a target sample count \(N_s\), total QMI run time is 41.52 + 2.04\(N_s\) µs.

IV. RAN POWER MODELS AND CELLULAR TARGETS

We now describe power modeling in RANs (§IV-A) and computational complexity of cellular networks (§IV-B).

A. POWER MODELING

RAN power models account for power by splitting the BS or CRAN functionality into the components and sub-components shown in Figs. 1 and 7. This section details these components and their associated power models. We follow the developments by Desset et al. [72] and Ge et al. [73].

1) RAN Power Model

A RAN BS (see Fig. 7) is comprised of a baseband unit (BBU), a radio unit (RU), power amplifiers (PAs), and a power system (PS). The entire BS power consumption \(P_{BS}\) is then:

\[
P_{BS} = \frac{P_{BBU} + P_{RU} + P_{PA}}{1 - \sigma_{AC}(1 - \sigma_{MS})(1 - \sigma_{DC})},
\]

where \(P_i\) is the ith BS component’s power consumption, and \(\sigma_{AC}(9\%), \sigma_{MS}(7\%), \text{ and } \sigma_{DC}(6\%)\) correspond to fractional losses of Active Cooling (A/C), Mains Supply (MS), and DC–DC conversions of the power system respectively [73].

The BBU performs the processing associated with digital baseband (BB), and control and transfer systems. The baseband includes computational tasks such as digital predistortion (DPD), up/down sampling or filtering, OFDM–FFT processing, frequency domain (FD) mapping/demapping and equalization, and forward error correction (FEC). The control system undertakes the platform control processing (PCP), and the transfer system processes the eCPRI transport layer. The total BBU power consumption \(P_{BBU}\) is then [72], [73]:

\[
P_{BBU} = P_{DPD} + P_{\text{Filter}} + P_{\text{FFT}} + P_{\text{FDsa}} + P_{\text{FDsb}} + P_{\text{FEC}} + P_{\text{PCP}} + P_{\text{CPRI}} + P_{\text{Leak}},
\]

where \(P_i\) is the ith BBU task’s power consumption, and \(P_{Leak}\) is the leakage power resulted from the employed hardware in processing these tasks. FD processing is split into two parts, with linear and non-linear scaling over number of antennas.
Where $RRHs$ perform low Layer 1 baseband processing, such as cyclic prefix removal and FFT-specific computation. The RU performs analog RF signal processing, consisting of clock generation, low-noise and variable gain amplification, IQ modulation, mixing, buffering, pre-driving, and analog–digital conversions. RU power consumption ($P_{RU}$) scales proportionally with number of transceiver chains, and each chain consumes about 10.8 W power [72]. For macro-cell BSs, each PA is typically consumes 102.6 W power [73].

### V. QA RESOURCE ESTIMATION

In this section, we estimate QA qubit count and their connectivity requirements that meet the cellular computational targets described above (§IV). While we exemplify this analysis from today’s 4G/5G perspective, same ideas can be used to study NextG systems as well.

#### A. QUBIT COUNT REQUIREMENT

To estimate qubit count, our approach considers the computational complexity of baseband tasks, their QUBO forms’ variable count, and run time on a QA device implementation. In particular, we convert the target TOPS complexity values (Table 3) into target problems per second (PPS), then estimate the qubit count required to achieve this PPS by analyzing QUBO forms of individual baseband computational tasks. We formulate the qubit count requirement as:

$$N_Q = \sum_k N_{Q,k}$$

$$N_{Q,k} = PPS_k \times N_{Q,p,k} \times T_{p,k}$$

$$PPS_k = \text{TOPS}_k / \text{Operations per problem}$$

### Table 3: Baseband unit’s computational complexity in Large MIMO base stations. Time and frequency duty cycles are at 100%, modulation is 64-QAM, and coding rate is 0.5. Values are in Tera operations per second. See §IV for abbreviations.

<table>
<thead>
<tr>
<th>BBU Task</th>
<th>Reference $N_A = 1$</th>
<th>4G (B/W = 20 MHz)</th>
<th>5G (B/W = 200 MHz)</th>
<th>5G (B/W = 400 MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_A = 2$</td>
<td>$N_A = 4$</td>
<td>$N_A = 8$</td>
<td>$N_A = 32$</td>
</tr>
<tr>
<td>$N_A = 64$</td>
<td>$N_A = 128$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPD</td>
<td>0.160</td>
<td>0.320</td>
<td>0.640</td>
<td>1.280</td>
</tr>
<tr>
<td>Filter</td>
<td>0.400</td>
<td>0.800</td>
<td>1.600</td>
<td>3.200</td>
</tr>
<tr>
<td>FFT</td>
<td>0.160</td>
<td>0.320</td>
<td>0.640</td>
<td>1.280</td>
</tr>
<tr>
<td>$FD_{lin}$</td>
<td>0.090</td>
<td>0.180</td>
<td>0.360</td>
<td>0.720</td>
</tr>
<tr>
<td>$FD_{dl}$</td>
<td>0.030</td>
<td>0.120</td>
<td>0.240</td>
<td>0.480</td>
</tr>
<tr>
<td>FEC</td>
<td>0.140</td>
<td>0.280</td>
<td>0.560</td>
<td>1.120</td>
</tr>
<tr>
<td>CPRI</td>
<td>0.720</td>
<td>1.440</td>
<td>2.880</td>
<td>5.760</td>
</tr>
<tr>
<td>PCP</td>
<td>0.400</td>
<td>0.800</td>
<td>1.600</td>
<td>3.200</td>
</tr>
<tr>
<td>Total</td>
<td>2.100</td>
<td>3.400</td>
<td>7.040</td>
<td>15.040</td>
</tr>
<tr>
<td></td>
<td>716.8</td>
<td>2,048.0</td>
<td>6,533.6</td>
<td>14,208.8</td>
</tr>
<tr>
<td></td>
<td>4,070.4</td>
<td>13,056.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[72], [73]. The RU performs analog RF signal processing, and it depends on parameters such as the bandwidth (B/W), modulation (M), coding rate (R), number of antennas ($N_A$), and time ($dt$) and frequency ($df$) domain duty cycles. Prior work [72], [73] present these TOPS complexity values for individual BBU tasks in a reference scenario (B/W = 20 MHz, M = 6, R = 1, $N_A = 1$, $dt = df = 100\%$), which we replicate in Table 3 as Reference. The scaling of these values follow [72], [73]:

$$\text{TOPS}_{\text{target}} = \text{TOPS}_{\text{ref}} \prod_k \left( \frac{X_{\text{target}}}{X_{\text{ref}}} \right)^{a_k}$$

where $X \in \{B/W, M, R, N_A, dt, df\}$ and $k \in [1,6]$ respectively. The scaling exponents $\{s_1, s_2, s_3, s_4, s_5, s_6\}$ are $\{1,0,1,1,0,1\}$ for DPD, Filter, and FFT, $\{1,0,0,1,1,1\}$ for $FD_{lin}$, $\{1,0,0,2,1,1\}$ for CPRI and FEC, and $\{0,0,0,1,0,0\}$ for PCP. These exponents are determined based on the dependence of BBU operation with the corresponding parameters [72], [73]. Table 3 reports the TOPS complexity values for representative 4G and 5G Large MIMO scenarios.
where \(N_Q\) is the total number of qubits the QA requires for the entire baseband processing, and \(N_{Q,k}\) is the qubit requirement for the \(k^{th}\) baseband task. \(PPS_k\) is the target problems per second, \(N_{Q,p,k}\) is the number of qubits per problem, and \(T_{p,k}\) is the run time per problem, of the \(k^{th}\) baseband task. We next demonstrate how to compute these values for \(FD_{al}\) and FEC tasks with running examples.

The \(FD_{al}\) task corresponds to the MIMO detection problem whose objective is to \textit{demodulate} the received wireless data into bits [79]. In a multi-user system with multiple antennas at the BS, the optimal MIMO detection performance is obtained by solving the QUBO objective function [23]:

\[
\arg\min_{\mathbf{x} \in \mathbb{C}^{N_r \times 1}} \left\| \mathbf{y} - \mathbf{H} \mathbf{x} \right\|^2
\]

(14)

where \(\mathbf{y} \in \mathbb{C}^{N_r \times 1}\) is received data, \(\mathbf{H} \in \mathbb{C}^{N_r \times N_t}\) is wireless channel, and \(\mathbf{x} \in \mathbb{C}^{N_t \times 1}\) is transmitted data to be estimated. \(N_t\) and \(N_r\) are the number of transmitters (users) and receivers (antennas) in the system respectively. We observe that upon expansion Eq. 14 becomes a quadratic minimization function, if each entry in \(\mathbf{x}\) is formulated as a linear function of variables. The search is over all possible \(\mathbf{x}\), and the entries in \(\mathbf{x}\) are selected based on the employed modulation scheme. For instance in BPSK modulation, we must search for values in \([\pm 1]\) and so each entry in \(\mathbf{x}\) takes the form \(2q - 1\), where \(q\) is a binary variable. Such formulations exist for various modulations (see [23] for details).

Solving a demodulation problem with \(Z\) users and \(Z\) antennas via state-of-the-art \textit{sphere decoding} algorithm requires on average 80 \((Z/64)^2\) million operations [80].\(^5\) Solving the same problem using QA requires \(N_{bps} \times Z\) qubits, where \(N_{bps}\) is the number of bits per symbol in the employed modulation scheme (see [23]). Therefore, for a typical 5G scenario: \(Z = 64\) and 64-QAM modulation \((N_{bps} = 6)\), we note that \(PPS_{FD_{al}}\) is 30.72M (i.e., 2457.6 TOPS/80M, see Table 3), \(N_{Q,p,FD_{al}}\) is 384 qubits, and \(T_{p,FD_{al}}\) is 41.52 + 2.04\(N_s\) \(\mu s\) \((\text{III})\). Substituting these values in Eq. 12 shows that the 5G \(FD_{al}\) processing requires 971K qubits with \(N_s = 20\) samples.

The FEC task corresponds to the channel \textit{decoding} problem which aims to correct the bit errors that noise and interference of the wireless channel inevitably introduce into the user data. In our analysis, we consider Low Density Parity Check (LDPC) codes employed in the 5G-NR traffic channel for FEC evaluation [81]. An \((M, N)\)-LDPC code is characterized by a binary-valued parity check matrix \([h_{ij}]_{M \times N}\), where each row defines a \textit{check constraint} and each column defines which check constraint a bit participates in. In particular, an entry \(h_{ij} = 1\) indicates that \(i^{th}\) bit participates in \(j^{th}\) check constraint. A check constraint is said to be satisfied when its modulo two bit-sum is zero \((i.e., \text{zero checksum})\), and a successful decoding occurs when all the check constraints of the code are satisfied. The optimal LDPC decoding performance

\(^5\)A 64 \(\times\) 64 MIMO detection problem requires 80 million operations [80], and it scales quadratic with number of antennas [72], [73].

Figure 8: QA qubit requirement at various problem run times to achieve spectral efficiency equal to CMOS processing, in a 5G scenario with 400 MHz BW and 64 antennas.

is obtained by solving the QUBO objective function [20]:

\[
\arg\min_q \left\{ W_1 \sum_{\forall c} L_{sat}(c) + W_2 \sum_{\forall j} \Delta_j \right\}
\]

(15)

where \(L_{sat}\) and \(\Delta\) are cost penalty functions, and \(W_1\) and \(W_2\) are positive weights. The function \(L_{sat}(c)\) takes the form: \((f(q) - 2f(a))^2\), where \(f(q)\) is the sum of solution variables participating in a check constraint \(c\), and \(2f(a)\) is a binary encoding of even integers via ancillary variables. The function \(\Delta_j\) takes the form: \((q_j - Pr(q_j = 1))^2\), computing the distance of a decoding candidate to the received data, where the probability \(Pr(q_j = 1)\) can be computed based on the received data for various modulations and channels [20].

The global minimum of this QUBO is a successful decoding that is most proximal to the received data (see [20] for details).

Solving an \((M, N)\)-LDPC decoding problem via state-of-the-art belief propagation algorithm requires \(N + 3w_r^2M - w_rM + 2w_c^2N + 4w_cN\) operations per iteration [82], where \(w_r\) and \(w_c\) are the average row and column weights of the parity check matrix respectively. Solving the same problem using QA requires \(N + Mt\) qubits, where \(t = \arg \min_{n \in \mathbb{Z}[2n+1]} \{2 \geq w_r - (w_r \mod 2)\}\) — see Ref. [20] for a full derivation. For the longest LDPC code in 5G: \(M = 4224\), \(N = 8448\), \(w_r = 8.64\), \(w_c = 20\), we note that \(PPS_{FEC}\) is 600K (i.e., 89.6 TOPS/150M, see Table 3) for typical 20 decoding iterations, \(N_{Q,p,FEC}\) is 21,120 qubits, and \(T_{p,FEC}\) is 41.52 + 2.04\(N_s\) \(\mu s\) \((\text{III})\). Substituting these values in Eq. 12 shows that the 5G FEC processing requires 1.04M qubits with \(N_s = 20\) samples.

\(FD_{al}\) and FEC tasks correspond to 75% of the 5G BBU’s baseband computation load. For the remaining 25% load, we project a proportionate qubit requirement. Fig. 8 shows the QA qubit requirement to satisfy the 5G baseband computational demand as a function of problem run time and sample count. Looking at Eq. 12 and Fig. 8, we see that for a given network operation scenario \((i.e., \text{fixed } PPS_k \text{ and } N_{Q,p,k})\), the problem run time \((41.52 + 2.04N_s)\) and sample count \((N_s)\) scale linearly with qubit requirement to achieve spectral efficiency equal to CMOS. In the figure, the sample count
indicates the required QA target fidelity in terms of error performance—when $N_c$ is 20, QA must reach ground state of the input problem in 20 anneal trials. Hence, QA must meet these run time–qubit count combinations to achieve spectral efficiency equal to CMOS. While in Fig. 8 we demonstrate an example scenario, a similar methodology is applied to estimate network-specific qubit requirements. Fig. 13 shows this qubit requirement for various bandwidths and antenna count choices (later described in §VII).

B. QUBIT CONNECTIVITY REQUIREMENT
To estimate qubit connectivity, we now analyze the native problem connectivity of the QUBOs described above. In this work, we consider future QA qubit connectivity to match the problem connectivity, which is typically challenging to realize for dense problems from a hardware perspective but will result in a highly efficient embedding process. Nevertheless, we describe promising methods that circumvent this issue.

From Eq. 14, we observe that the connectivity graph of the $F_{D_{al}}$ task is a complete graph on $N_{bpe}$ × $Z$ variables. For a typical 5G scenario: $N_{bpe} = 6$ and $Z = 64$, this corresponds to a 384-qubit full connectivity, which is challenging to realize on QA devices. Scaling to more users and higher modulation schemes envisioned in NextG will increase this qubit connectivity requirement even further, making it more challenging from a hardware perspective. To address this connectivity issue, hybrid QPU–CPU approaches that decompose a large QUBO into a number of smaller sub-QUBOs realizable on hardware may be necessary. Existing methods such as that of $gbsolv$ based on Glover’s algorithm provides such a hybrid interface for generic problems, rendering it useful in this regard [83]–[85]. Further decomposition approaches tailored to the $F_{D_{al}}$ task also exist, which demonstrate that decomposed sub-problems can be parallelized via warm state initialization to obtain good performance [86]. While the size of decomposed sub-problems can be chosen flexibly, we note that such decomposition methods typically entail performance loss due to reduced complexity. Quantifying this performance loss for NextG problems requires an empirical evaluation on future QAs—we leave this for future work. Nevertheless, this loss is observed to be negligible for 5G problems [86].

Unlike the $F_{D_{al}}$ task, the connectivity graph of the FEC parity task is highly sparse due to the inherent nature of LDPC codes being low density codes. Each qubit in LDPC decoding typically requires a different connectivity degree, which can be precisely calculated as follows. Consider an LDPC code with $M$ rows and $N$ columns in its parity check matrix, and let $rw_i$ be its $i^{th}$ row’s weight (i.e., number of 1s in $i^{th}$ row). Compute $t_i = \arg \min_{n \in \mathbb{Z}} \left\{ 2^n+1 - 2 \geq rw_i - (rw_i \mod 2) \right\}$, the number of ancillary qubits required for $i^{th}$ row. Then the connectivity degree required for $t_i$ ancillary qubits is $rw_i + t_i - 1$ for all $i \in [1, M]$. Each ancillary qubit is unique, and so the number of qubits whose connectivity degree we have determined above is $\sum_{i=1}^{M} t_i$. Alongside ancillary qubits, the decoding requires $N$ distinct solution qubits whose connectivity degree is calculated next. To compute $j^{th}$ solution qubit’s connectivity degree, construct a submatrix of the parity check matrix by eliminating the rows whose $j^{th}$ column entry is zero. Then compute $C_j$, the number of non-zero columns in this submatrix. The connectivity degree required for the $j^{th}$ solution qubit is then $(\sum_{i=1}^{N} h_{ij}) + C_j - 1$ for all $j \in [1, N]$, where $h_{ij}$ is the $(i, j)^{th}$ entry of parity check matrix. These connectivity degrees are derived by analyzing the QUBO form given in Eq. 15 (see Ref. [20]). For decoding the longest LDPC code in 5G, QA needs 21,120 qubits, where {46%, 28%, 2%, 11%, 13%} of the qubits require {<10, 11–30, 30–60, 60–100, >100} couplers per qubit respectively. The highest connectivity degree is 205, and the average connectivity degree is 34.28. While we present numbers for the longest LDPC code, a similar methodology can be used to compute connectivity degree requirement for smaller LDPC codes in practice. Further, all the quadratic coefficients of the LDPC QUBO function remain constant for a given a parity check matrix (i.e., only linear coefficients change from problem to problem), which eases the coupler programming process, making it a favorable candidate for a tailored hardware design.

VI. EVALUATION: POWER AND COST ANALYSIS
This section presents a holistic power and cost comparison between QA and CMOS in cellular wireless networks. Our methodology compares CMOS and QA processing at equal spectral efficiency outcomes. We specify the same BBU targets (Table 3) with CMOS and QA hardware, ensuring equal bits processed per second per Hz per km$^2$.

The power consumption of CMOS hardware depends on its performance-per-watt efficiency and the amount of computation at hand. Technology scaling improves this efficiency from generation to generation, inversely proportional to the square of its transistors’ core supply voltage ($V_{dd}$) [87]. A 65 nm CMOS device ($V_{dd} = 1.1$ V) has a 0.04 TOPS/Watt efficiency, from which we compute the same for today’s 14 nm CMOS ($V_{dd} = 0.8$ V) and future 1.5 nm CMOS ($V_{dd} = 0.4$ V), via $V_{dd}^2$ scaling, and they obtain a 0.076 and 0.3 TOPS/Watt efficiency respectively [11], [72], [88]. Using this hardware efficiency and the TOPS requirements of Table 3, we compute CMOS hardware power consumption. Additional power results from leakage currents in CMOS transistor channel, and this leakage power is set to 30% of dynamic power [72].

Power consumption of D-Wave’s QA is ca. 25 kW, dominated by its refrigeration unit (see Supplementary information–[32]). Additional power draw due to the computation at hand is negligible compared to QA refrigeration power, since the QPU resources used for computation are thermally isolated in a superconducting environment. This power requirement is further not expected to significantly scale up with increased qubit numbers [32], [34], due to the fairly constant power consumption of pulse-tube dilution refrigerators which are used to cool the QPU in practice [32], [57], [89]. More general NISQ processors such as Google’s Sycamore (see Supplementary information–[33]) and IBM’s Rochester [90] also show a similar ca. 25 kW power consumption and a fairly
constant scaling with increased qubit numbers [34]. However, to maintain this 25 kW power for the entire 5G baseband processing, sufficient amount of qubits are required, all under the same refrigeration unit (couplers do not require additional space [35], [36]). This raises the question—how many qubits are possible in a QA refrigeration unit? To answer this question, we consider the physical size of qubits in their unit cell packaging (a die) versus the available space in the dilution refrigerator. The number of useful square dies \( N_d \) of length \( L_d \) placed onto a wafer of radius \( R_w \) is approximately [91]:

\[
N_d = \frac{\pi R_w^2}{L_d^2} - \frac{1}{16} \log_2 R_w.
\]

A square die of eight qubits requires \( 335 \times 335 \mu m^2 \) QPU chip area with \( L_d = 335 \mu m \), and a dilution refrigerator’s experimental space has a radius \( R_w = 250 mm \) [57]. Substituting these values in the above equation gives \( N_d \approx 1.75M \), which implies \( \approx 14 \) million qubits allowed in a refrigeration unit. Larger dilution refrigerators such as IBM’s Goldeneye can accomodate at least 6× qubits than a regular dilution refrigerator considered above [92]. Since qubit count estimates for 5G (cf. §V, §VII) are well below this allowed limit, QA power consumption is 25 kW for 5G baseband processing.

### A. BS AND CRAN POWER COMPARISON

Applying the foregoing power analysis, Fig. 9 reports power consumption results of 4G and 5G Large MIMO BSs where one antenna at the BS serves one user. In Fig. 9(a), we see that the power amplifier (PA) is the dominating component of 4G BS power consumption, accounting for 57–58% of the total BS power, as identified in several prior works [72], [73], [77]. But, as the network scales to higher bandwidth and antennas envisioned in 5G, the BBU becomes the dominant power consuming component (see Fig. 9(b)), accounting for 69–74% of the total BS power. This quick escalation in power from 0.35–1.43 kW in 4G to 34.7–261.3 kW in 5G is mainly due to the non-linear FD processing (§IV-A), and the increased network bandwidth consequence of millimeter-wave communication. Fig. 10(a) reports the power consumption results of 4G BS, where QA is used for BBU’s baseband processing. In comparison to CMOS—Fig. 9(b), QA reduces BS power by 41 kW and 188 kW in 64 and 128 antenna systems. Fig. 10(b) shows power consumption in a CRAN setting with three 64-antenna BSs, where the fronthaul is allowed a 100 Gbps bandwidth. In comparison to CMOS, QA processing reduces CRAN power by 159 kW (55% lower).
B. BASEBAND POWER COMPARISON

This section compares the power consumption of QA and CMOS for the BBU’s baseband processing along a variety of base station (Fig. 11) and CRAN (Fig. 12) operation scenarios.

1) Base Station

**Across MIMO degree.** MIMO degree (·) is the number of antennas used to serve one user, where MIMO(8) and MIMO(4) are the status quo 5G implementations and MIMO(1), referred to as Large MIMO, is the ideal scenario that maximizes spectral efficiency. Figures 11(a) and 11(d) report these results, showing that with MIMO(8), both 14 and 1.5 nm CMOS processing require lesser power than QA, at all bandwidths and antenna counts. However, as we decrease the MIMO degree to MIMO(1), we observe that QA achieves power advantage over 14 nm CMOS (Fig. 11(a)) in 256-antenna systems at all bandwidths and in 64-antenna systems at 200MHz and 400MHz bandwidths. QA processing at 100, 200, and 400 MHz bandwidth 5G BSs with 256 antennas benefit in power over 1.5 nm CMOS (Fig. 11(d)).

**Across antenna count.** Figs. 11(b) and 11(e) compare power consumption of BSs at various antenna count choices. In 32-antenna BSs at 200 and 400 MHz bandwidths, we note that the power consumption of both 14 and 1.5 nm CMOS is lesser than QA at all MIMO degrees. This is because when the antenna count is low, the number of users supported at the BS and their resulting computational demand is low, leading to low CMOS power consumption. However, as we increase the antenna counts we see a significant rise in CMOS power consumption. In 256-antenna systems with 200 and 400 MHz bandwidths, QA benefits in power over 14 nm CMOS at MIMO degrees 4, 2, and 1. In comparison to 1.5 nm CMOS, the same systems benefit in power at MIMO degrees 2 and 1.

**Across network bandwidth.** From Figs. 11(c) and 11(f), we see that the lowest bandwidth for which QA achieves power advantage over 14 nm CMOS are 20 MHz bandwidth 256-antenna systems (Point ‘A’), 50 MHz bandwidth 128-antenna systems (Point ‘B’), and 160 MHz bandwidth 64-antenna systems (Point ‘C’) systems. In comparison to 1.5 nm CMOS, such points correspond to 60 MHz bandwidth 256-antenna systems (Point ‘D’), and 190 MHz bandwidth 128-antenna systems (Point ‘E’) systems.

2) CRAN

**Massive MIMO.** Fig. 12(a) compares power consumption of 1.5 nm CMOS against QA in a CRAN setting with 2–5 Massive MIMO(4) base stations. We see that even when CRAN handles five 64-antenna base stations, power consumption of 1.5 nm CMOS is lesser than QA at all bandwidths. Whereas a CRAN handling more than one 256-antenna 400 MHz base stations benefits in power with QA over 1.5 nm CMOS.
Further, CRAN with at least four 256-antenna 200MHz base stations requires lesser power with QA than 1.5 nm CMOS.

**Large MIMO.** Fig. 12(b) investigates how power consumption of 1.5 nm CMOS compares with that of QA when CRAN handles Large MIMO base stations, whose MIMO degree is one. In a CRAN setting with 2–5 256-antenna BSs, QA requires 1–2 orders of magnitude lesser power than 1.5 nm CMOS. With at least two 400 MHz bandwidth and four 200 MHz bandwidth 64-antenna base stations, CRAN achieves a power advantage with QA over 1.5 nm CMOS.

**Cost and Carbon savings.** In Fig. 12(c), we see the summary of OpEx cost savings and carbon emission reductions associated with the respective power savings, computed by considering an average $0.143 (USD) electricity price and 0.92 pounds of CO$_2$ equivalent emitted per kWh [93], [94]. The figure reports the savings of QA against 1.5 nm CMOS in a CRAN setting with 400 MHz bandwidth 256-antenna base stations in Massive MIMO(4) and Large MIMO scenarios. To provide a cost and carbon benefit over CMOS hardware, assuming CMOS CapEx is negligible, future QAs’ CapEx must be lower than the respective OpEx savings. For instance, if QA was to be employed in a CRAN setting with five Large MIMO base stations, a QA CapEx lower than 33K qubits is required to achieve equal spectral efficiency to CMOS, and this qubit requirement is projected to become available by the year 2026 based on current industry trends (Fig. 14). However, leveraging QA for such a system does not provide power advantage in comparison to both 14 nm and 1.5 nm CMOS devices (see Figs. 11(c), 11(f)).

**Roadmap for feasibility.** The processing of a base station with 10-MHz bandwidth and 32 antennas requires 33K qubits in the QA hardware for QA to achieve equal spectral efficiency to CMOS, and this qubit requirement is projected to become available by the year 2026 based on current industry trends (Fig. 14). However, leveraging QA for such a system does not provide power advantage in comparison to both 14 nm and 1.5 nm CMOS devices (see Figs. 11(c), 11(f)).

**Roadmap for power dominance.** From Figs. 11(c) and 11(f), we note that Points A–E are the lowest bandwidths at each antenna count for which QA achieves power advantage over CMOS. Fig. 13 shows the number of qubits required in the QA hardware to process these systems (Points A–E) with equal spectral efficiency to CMOS. The figure shows that to achieve a power dominance over 14 nm CMOS, at least 537K qubits (Point ‘A’) are required in the QA hardware, and this qubit requirement is projected to become available by the year 2034 (Fig. 14). QA with at least 1.6M qubits benefit in power over 1.5 nm CMOS, and such a QA is predicted to become available by the year 2037 (Fig. 14). In summary, our analyses show that power advantage of QA over CMOS is a predicted 11–14 years away. Fig. 14 summarizes Fig. 13 in a theoretical perspective.
feasibility timeline, showing the years by which QA enables these base station operation scenarios along with associated power advantage/loss.

VIII. CONCLUSION

This paper makes the case for the future feasibility of QA processing-based wireless networks from a cost/power perspective. Our extensive analysis of QA technology projects quantitative targets that future QAs must meet in order to provide benefits over CMOS in terms of performance, power, and cost. Our results show that with QA hardware advancements, a cost/power benefit of QA over CMOS is a predicted 11–14 years away. Furthermore, fundamental physical advances in the QA technology itself, which we do not leverage in the projections given in this paper, may offer further benefits, advantaging our projected timelines. Examples of these advances include faster annealing times (<40 ns) and/or qubits with longer coherence lifetimes (such as the qubits in IARPA’s QEO and DARPA’s QAFS QA chips [96]) that enable coherent quantum annealing regimes, benefiting future QA spectral efficiency [61], [97]. While we acknowledge the practical feasibility of QA processors to be at least tens of years away, this early study informs NextG QA hardware design and wireless networks.

Limitations of this study. We stress that our analysis assumes that QA devices will continue to advance according to the current industry trends, and that any future technological breakthrough or setback is not accounted for. In such an event, our projections must be revised accordingly, nevertheless, the methodology remains the same.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1824357. P.A.W. is supported by the Engineering and Physical Sciences Research Council (EPSRC) Hub in Quantum Computing and Simulation, Grant Ref. EP/T001062/1. K.J. and S.K. gratefully acknowledge a gift from InterDigital Corporation. We thank Andrew J. Berkley, Keith Briggs, Andrew D. King, Catherine McGeoch, Davide Venturelli, and Catherine White for useful discussions.

References

SRIKAR KASI is Ph.D. student in the Department of Computer Science at Princeton University. His research interest is in wireless networks, quantum and quantum-inspired computing, graph theory, and mobile systems. He received B.Tech degree (2018) in Electrical Engineering from the Indian Institute of Technology Delhi, and M.A degree (2022) in Computer Science from the Princeton University. He is a recepient of Qualcomm Innovation Fellowship 2021 (North America).

PAUL WARBURTON received the BA degree in Electrical and Information Sciences in 1990 and the PhD degree in Materials Science in 1994, both at the University of Cambridge, UK. He is currently Professor of Nanoelectronics at University College London (UCL), UK. From 1994 to 1995, he was a post-doc at the University of Maryland, USA. From 1995 to 2001 he was Lecturer at King’s College London, UK. Since 2001 he has been at UCL where he holds a joint appointment between the London Centre for Nanotechnology and the Department of Electrical and Electronic Engineering. His research interests include superconducting devices, quantum annealing and nanofabrication.

JOHN KAEWELL joined InterDigital in 1986 where he has developed multiple generations of wireless communication systems. He leads InterDigital’s exploration of using quantum computing to solve wireless optimization problems and is working on applying Machine Learning to improve wireless system performance. Mr. Kaewell has been inducted into the Drexel College of Engineering Circle of Distinction and has received InterDigital’s Chairman award. He holds 56 US Patents and over 650 patents and applications worldwide.

KYLE JAMIESON is Professor of Computer Science and Associated Faculty in Electrical and Computer Engineering at Princeton University. His research focuses on mobile and wireless systems for sensing, localization, and communication, and on massively-parallel classical, quantum, and quantum-inspired computational structures for NextG wireless communications systems. He received the B.S. (Mathematics, Computer Science), M.Eng. (Computer Science and Engineering), and Ph.D. (Computer Science, 2008) degrees from the Massachusetts Institute of Technology. He then received a Starting Investigator fellowship from the European Research Council, a Google Faculty Research Award, and the ACM SIGMOBILE Early Career Award. He served as an Associate Editor of IEEE Transactions on Networking from 2018 to 2020. He is a Senior Member of the ACM and the IEEE.