Review of Scientific Instruments

RESEARCH ARTICLE | APRIL 26 2022

The QICK (Quantum Instrumentation Control Kit): Readout and control for qubits and detectors

Leandro Stefanazzi; Kenneth Treptow; Neal Wilcer; Chris Stoughton ; Collin Bradford ; Sho Uemura ; Silvia Zorzetti; Salvatore Montella; Gustavo Cancelo ; Sara Sussman ; Andrew Houck; Shefali Saxena ; Horacio Arnaldi ; Ankur Agrawal ; Helin Zhang; Chunyang Ding; David I. Schuster

Rev. Sci. Instrum. 93, 044709 (2022)
https://doi.org/10.1063/5.0076249

AIP Publishing

Review of Scientific Instruments ARTICLE scitation.org/journal/rsi

The QICK (Quantum Instrumentation Control Kit): Readout and control for qubits and detectors

Cite as: Rev. Sci. Instrum. 93, 044709 (2022); doi: 10.1063/5.0076249
Submitted: 22 October 2021 • Accepted: 15 March 2022 •
Published Online: 26 April 2022

Leandro Stefanazzi,[1] Kenneth Treptow,[1] Neal Wilcer,[1] Chris Stoughton,[1] Collin Bradford,[1] Sho Uemura,[1] Silvia Zorzetti,[1] Salvatore Montella,[1] Gustavo Cancelo,[1,a)] Sara Sussman,[2] Andrew Houck,[2] Shefali Saxena,[3] Horacio Arnaldi,[4] Ankur Agrawal,[5] Helin Zhang,[5] Chunyang Ding,[5] and David I. Schuster[5]

AFFILIATIONS

[1] Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA
[2] Department of Physics and Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08544, USA
[3] Argonne National Laboratory, Lemont, Illinois 60439, USA
[4] Instituto Balseiro, Bariloche, RN 8402, Argentina
[5] Department of Physics and Pritzker School of Molecular Engineering, University of Chicago, Chicago, Illinois 60637, USA

[a)] **Author to whom correspondence should be addressed:** cancelo@fnal.gov

ABSTRACT

We introduce a Xilinx RF System-on-Chip (RFSoC)-based qubit controller (called the Quantum Instrumentation Control Kit, or QICK for short), which supports the direct synthesis of control pulses with carrier frequencies of up to 6 GHz. The QICK can control multiple qubits or other quantum devices. The QICK consists of a digital board hosting an RFSoC field-programmable gate array, custom firmware, and software and an optional companion custom-designed analog front-end board. We characterize the analog performance of the system as well as its digital latency, important for quantum error correction and feedback protocols. We benchmark the controller by performing standard characterizations of a transmon qubit. We achieve an average gate fidelity of $\mathcal{F}_{avg} = 99.93\%$. All of the schematics, firmware, and software are open-source.

I. INTRODUCTION

Quantum computers are predicted to outperform classical computers in problem domains such as decryption,[1,2] secure communication,[3,4] quantum chemistry,[5] and machine learning.[6] Quantum bits (qubits) have been developed in several different platforms including trapped ions,[7] superconducting qubits,[8] semiconductor quantum dots,[9] color centers,[10] and neutral atoms.[11] In all of these systems, the ability to synthesize a large number of control signals, measure the states of the qubits, and perform feedback in real-time is a critical requirement. Historically, this has been done using either expensive general-purpose test equipment or proprietary embedded systems. This has made controlling even moderate size quantum processors with tens of qubits, such as those hosted by IBM or Google,[12,13] prohibitive for academic labs and small startups. We have developed a RF/Field-Programmable Gate Array (FPGA)-based system that is flexible enough to support different platforms and is affordable to academic laboratories.

There have been several platforms developed that integrate fast RF digital to analog converters (DACs) and analog to digital converters (ADCs) with Field-Programmable Gate Arrays (FPGAs). Early academic efforts in the superconducting qubit community were made at Delft for pulse routing[14] and at ETH Zurich for real-time processing of measurements.[15] The first demonstration of quantum error correction to show improvement over the physical constituents was enabled by an FPGA-controlled data acquisition system.[16] The ion trap community has developed the FPGA-based Advanced Real-Time Infrastructure for Quantum physics (ARTIQ) system[17] for control of their systems, albeit on slower timescales than those required for solid-state qubits, which typically have faster gate speeds (and decoherence rates). Commercial products with

FPGA-enabled real-time pulse synthesis and readout have recently become available from vendors such as BBN,[18] Keysight,[19] Zurich Instruments,[20] and Quantum Machines.[21] An open-source qubit controller was recently developed by LBNL and UC Berkeley.[22] These solutions all use a conventional method by which the FPGA-controlled DAC synthesizes the intermediate frequency (IF) envelope, which is upconverted via in-phase and quadrature (IQ) mixing with a local oscillator supplied by an analog RF source.

More recently, a newer generation of RF DACs operates at high enough sampling rates that it becomes possible to directly synthesize microwave pulses without any kind of upconversion,[23] eliminating the need for meticulous calibration.[24] In this work, we take advantage of a newly developed FPGA platform, the RFSoC from Xilinx,[25] which integrates high-speed DACs, ADCs, programmable FPGA logic, and a conventional microprocessor in the same package. RFSoC-based qubit controllers have been homemade by several academic labs,[26,27] and one was made available commercially by Intermodulation Products.[28]

The Quantum Instrumentation Control Kit (QICK) is an open-source RFSOC-based platform, which combines FPGA firmware for real-time processing and a Python interface running the PYNQ[29] operating system. The initial implementation of the QICK uses the ZCU111[30] evaluation board with eight DACs and ADCs. This allows for pulse synthesis up to 6 GHz directly using the XM500 RFMC balun card provided in the evaluation kit and higher by using either external sources and mixers or the custom RF board we have developed. The system can synthesize and digitally upconvert arbitrary pulses, measure and digitally downconvert incoming signals, and perform real-time decisions and feedback based on the input. The system is open-source[31] and can run on commercially available hardware at a price significantly lower than commercial offerings at this time.

The rest of the paper is organized as follows: In Sec. II, we describe the hardware and its capabilities. In Sec. III, we describe the controller architecture and the microsequencer with precision timing blocks along with the digital upconversion and downconversion chains. Next, in Sec. IV, we characterize the analog performance of the direct output of the ZCU111 using both the XM500 RFMC balun card and the custom analog QICK RF board. We conclude Sec. IV by demonstrating the use of the system to fully characterize a superconducting qubit previously used to perform a qubit-enhanced search for dark matter.[32] The custom RF board has not yet been used in qubit experiments due to part shortages and supply chain delays. Instead, discrete RF hardware was used as shown in Sec. IV B, Fig. 13.

II. DIGITAL AND ANALOG HARDWARE

The QICK (Quantum Instrumentation Control Kit) is a multi-input, multi-output, high performance controller designed for qubit systems and superconducting detectors. The QICK can be used as a flexible instrument to control and characterize new qubits and detectors, or it can be used as a module in a multi-module architecture for a large detector instrument or quantum computer. A precursor to the QICK is the FNAL-Gen2 fMESSI system, which generated two chronograph instruments: DARKNESS and MKID (Microwave Kinetic Inductance Detector) Exoplanet Camera. MEC uses a stack-up of 20 fMESSI boards, and it has been operating at the 8 m Subaru telescope since 2018.[33] The QICK is designed to be self-contained: the user directly connects RF lines between the QICK and the dilution refrigerator containing the qubits. With appropriate firmware, the QICK hardware can also be used for superconducting detectors such as MKIDs (Microwave Kinetic Inductance Detectors) or TESes (Transition Edge Sensors) connected to RF microchips for high-density frequency multiplexing. Since the RFSoC analog inputs each have 2 GHz of continuous useable bandwidth, the number of detectors per channel depends on the channel bandwidth and separation. A typical application[34] has 1000 MKID channels separated by 2 MHz. The QICK could be used to read out several thousands of multiplexed detectors, a number that will ultimately be limited by the available FPGA logic in the specific design. MKIDs and TESes coupled to microwave cavities require an excitation tone at the resonance frequency of the pixel, and so, the QICK DACs will be used

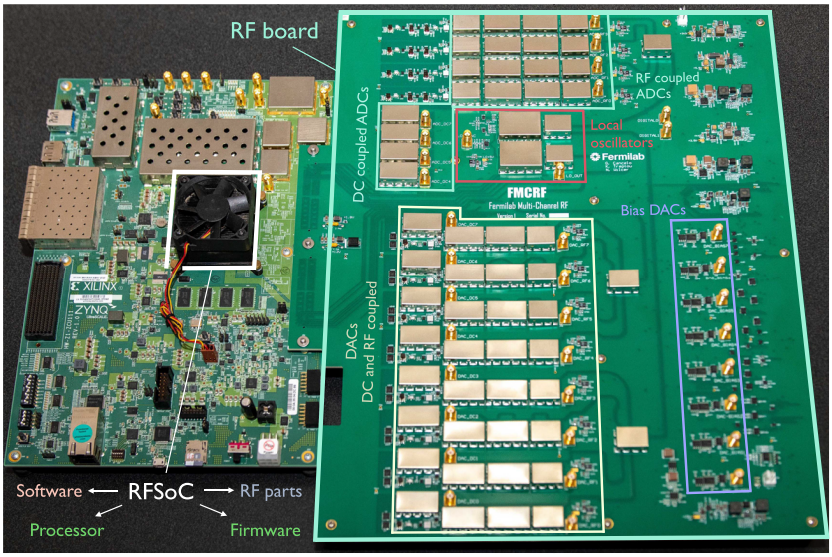

FIG. 1. The Quantum Instrumentation Control Kit (QICK). The QICK consists of two pieces of hardware: the commercial ZCU111 RFSoC evaluation board (left), which connects to the QICK RF board (right), which can be used for additional signal up/downconversion, amplification, and filtering.

to provide the same number of tone excitations. Using the QICK as a MKID/TES controller will be described in a separate paper.

The QICK system is simpler, more compact, and cost-effective than the conventional qubit control stack that has many auxiliary components (e.g., two DACs and one external local oscillator per channel). The QICK hardware consists of two parts, a main evaluation board and an optional custom-designed analog front-end shown together in Fig. 1.

A. Xilinx ZCU111 RFSoC evaluation board

The QICK takes advantage of the highly integrated RFSoC FPGA. The XCZU28DR RFSoC chip (Fig. 2) has eight 6.5 GS/s digital to analog converters (DACs) and eight 4 GS/s analog to digital converters (ADCs). Both the DAC and ADC blocks include configurable IQ digital up/down conversion, an integrated numerically controlled oscillator (NCO), a gain matrix, and digital filters with interpolation/decimation. They are easily integrated to the logic through standard Advanced eXtensible Interface (AXI) interfaces and avoid the use of high-power drivers needed with external A/D and D/A devices that require Low Voltage Differential Signal (LVDS) or JEDEC (Joint Electron Devices Engineering Council protocol) interfaces. The RFSoC also contains several different antiradiation missile Advanced Reduced Instruction Set Computer (RISC) Machines (ARM) processors and 70 MB of memory and multiple interfaces.

Most commercial qubit controllers have DACs with <1 GHz of analog bandwidth, so RF qubit control pulses (typically 4–6 GHz) must be upconverted with analog mixers. In contrast, the qubit controller presented here can directly synthesize carrier frequencies of up to 3 GHz in first Nyquist zone mode, and it can directly synthesize carrier frequencies of up to 6 GHz in second Nyquist zone mode. This eliminates mixer spurs driving undesirable transitions and eliminates the need to carefully calibrate IQ mixer offsets and gains.

B. Analog front-end

The QICK RF board contains more than 200 components, including amplifiers, mixers, filters, local oscillator generators, switches, and drivers. All RF and DC coupled outputs/inputs are accessible via Sub-Miniature Version A (SMA) connectors. The QICK only requires a 50 W, 12 V DC power supply. In the following sections, we describe the integrated RF/FPGA chip at the center of the controller, the RF board's RF and DC coupled outputs and inputs, and the RF board's components, which are used for bias and digital input–output (I/O).

The QICK has an integrated low jitter master clock; however, a stack-up of multiple QICK boards can be synchronized to a single external stable reference. See Appendix A for details of clock synchronization measurements. Figure 3 shows the high-level block diagram of the controller, which will be described in detail in Sec. III.

The custom RF board (larger board on the right side of Fig. 1) extends the eight RFSoC DAC outputs to either RF or DC coupled amplification and filtering. A simplified block diagram of the output chain is shown in Fig. 4. Every DAC output is connected to a software-controllable switch that allows the user to choose between an RF output or a DC coupled output. The RF output path

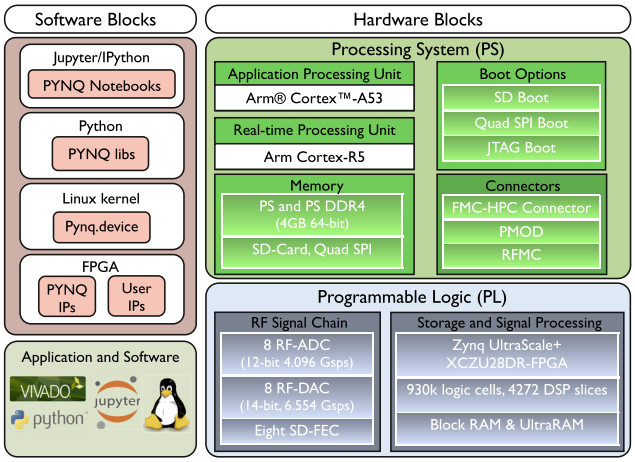

FIG. 2. The Xilinx XCZU28DR RFSoC chip block diagram. The chip consists of software blocks and hardware blocks. The software blocks can be subdivided into lower-level blocks such as the Linux kernel and higher-level blocks such as the PYNQ software library whose functions are called from Jupyter notebooks. The hardware blocks can be subdivided into blocks related to the processing system (PS) and the programmable logic (PL).

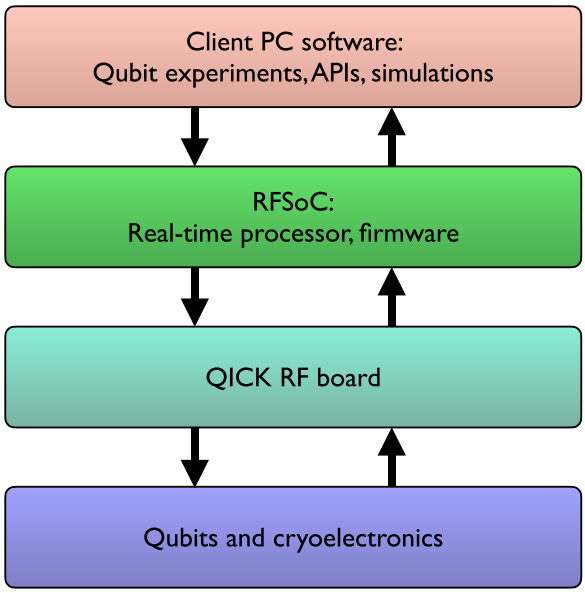

FIG. 3. The QICK stack-up. The user runs qubit experiments via the QICK Python API (top level of the stack). The qubit experiment is sent to the RFSoC (second level of the stack) and translated into FPGA-level instructions. Signals generated by the RFSOC are processed further by the QICK RF board (third level of the stack) and then sent to the qubits (the bottom level of the stack). Qubit measurements are then read into the QICK in the reverse order of the stack-up, and the measurement results are returned by the Python API. Up to an entire quantum program can be executed by the RFSoC without high-level software calls.

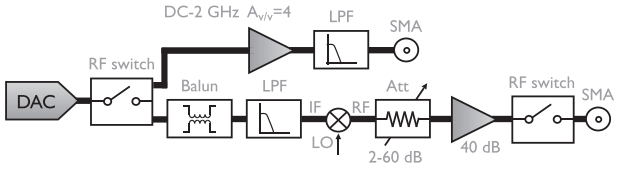

FIG. 4. The QICK RF board DAC output schematic. In software, the user can select the DAC output to be RF or DC coupled. The output signal is then processed by the associated DAC output chain. Note that the two step attenuators and three RF amplifiers are interleaved to optimize power linearity.

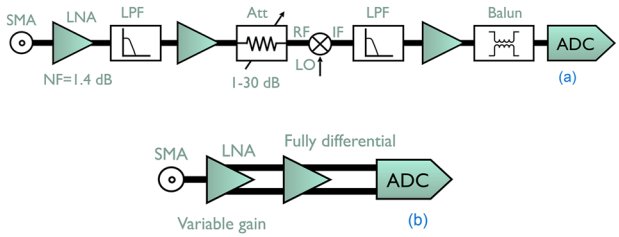

FIG. 6. The QICK RF board ADC input schematic. The QICK has four dedicated RF inputs (a) optimized for 4–8 GHz and four dedicated DC coupled inputs (b) with a bandwidth of 1.5 GHz.

upconverts the RFSoC DAC output signal using the MM1-0212HSM double balanced mixer, which operates between 2 and 12 GHz. The RF board's amplifying chain has been optimized to have the highest performance between 3 and 8 GHz (Fig. 5). The output of the mixer (typically ~ −23 dBm) is amplified by 40 dB and attenuated by two digital step attenuators. The step attenuators introduce a minimum insertion loss of 1 dB and a total maximum attenuation of up to 60 dB in 0.25 dB steps. Therefore, the RF output power dynamic range is 4 to −56 dBm. Alternatively, the DAC output can be switched to a DC coupled amplifier with 2 GHz of bandwidth. The main purpose of the DC coupled output is to control fast unmodulated signals such as fast flux pulses for fluxonium qubits as in Ref. 35 or for other quantum systems, which integrate fast voltage or flux pulses, such as spin qubits in quantum dots.[36]

Four of the eight QICK RF inputs (Fig. 6) are designed for RF signals, and the other four are DC coupled with an analog bandwidth of 1.5 GHz (for use as auxiliary oscilloscope or spectrum analyzer inputs, for example). The four RF inputs are designed to amplify low noise RF signals coming from the dilution refrigerator. The noise temperature of the RF input channel is governed by the noise figure of the first amplifier in the chain, the MACOM MAAL-011130, which is 1.4 dB. The input signal is amplified by four high-gain, high-P1dB compression amplifiers and attenuated by a step attenuator with a maximum of 30 dB in 0.25 dB steps. The RF input chain is mixed down using the same Marki mixer used for upconversion. Therefore, the RF input dynamic range is −60 to −90 dBm. Note that the 4 GHz analog bandwidth of the RF ADCs suggests that we can frequency multiplex the readout of several qubits onto the same line. If we were to have 100 MHz bandwidth readout cavities that were separated by 200 MHz, ten qubits could be simultaneously read out by the same RF ADC with room to spare.

The QICK RF board also includes additional DACs and digital I/O that are separate from the RFSoC. There are eight 20-bit DAC outputs for the biasing and DC control of flux for quantum devices that are tuned by the flux or voltage such as flux qubits or spin qubits in quantum dots. The bias maximum output voltage is ±10 V with 1 ppm resolution, 1 ppm integral nonlinearity, 7.8 nV/$\sqrt{\text{Hz}}$ of white Gaussian noise, and a 1/f noise knee at 4 Hz. There are also 16 digital software-configurable bidirectional I/O for the purpose of synchronization and triggering between the QICK and other instrumentation.

The on-board local oscillators (LOs) for the 12 RF mixers (eight upconverters and four downconverters) are generated by three LMX2595 phase locked loop (PLL) and fractional frequency synthesizers. Each PLL has two outputs, which are split in two by a Mini-Circuits EP2C+ splitter. Each PLL can be set to a different LO, allowing for two sets of four DACs and one set of four ADCs to have separate LO frequencies. Each on-board LO can be tuned from DC to 15 GHz.

The PLL output is amplified by an HMC788 to achieve optimum LO power at the mixers for a broadband range of ~1–10 GHz or more. The high frequency (10 kHz to 10 MHz) integrated jitter is 55 fs. The 10 Hz to 10 MHz integrated jitter is ~500 fs.

Appendix B shows a table with output and input bandwidths of the QICK system.

III. SYSTEM ARCHITECTURE AND FUNCTIONALITY

The functionality of the QICK system is divided between the Processing System (PS) and Programmable Logic (PL), shown in

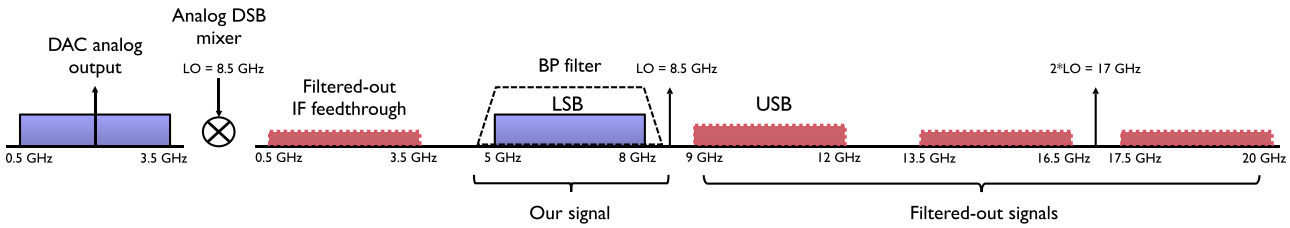

FIG. 5. The QICK RF board upconversion schematic. In this example, the QICK RF board LO is parked at 8.5 GHz. The user can place the IF signal anywhere in the 0.5–3.5 GHz band, and then, it is upconverted into the 5–8 GHz band. The QICK RF board filters the spurs in the unwanted sidebands. Although the fast digital DDS can generate a carrier lower than 0.5 GHz, a minimum of few hundred MHz is desired to avoid getting very close to the edge of the filter that suppresses the LO and higher frequency signals. The current version of the RF output includes a Mini-Circuits FCN-1800+ low-pass filter that can be replaced if more analog bandwidth is needed.

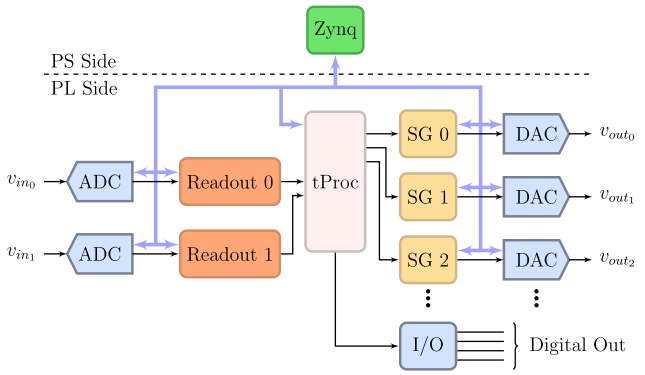

FIG. 7. The QICK firmware block diagram. The QICK firmware mainly consists of the timed-processor (tProcessor) block (Fig. 8), the signal generator (SG) blocks (Fig. 9), and the readout blocks (Fig. 10). Instructions are passed between the RFSoC Zynq processor and the firmware blocks. These instructions cause signals to be sent and received from the RF blocks (DAC blocks, ADC blocks, and digital output blocks).

Fig. 7. The PS of this UltraScale+ device is a ZYNQ system with its own DDR4 memory, which runs the Linux operating system on a multi-core ARM processor. The PS uses PYNQ libraries and drivers to enable direct memory access (DMA) to the PL. The user interface is typically via Jupyter notebooks accessed from a remote web browser. Firmware in the PL includes signal generator blocks and readout blocks controlled by a timed-processor (tProcessor) block to implement time-critical functions.

Data flow between the PS and PL uses AXI interfaces. Fast data transfers are implemented using DMA logic on the PL side. The firmware version has a companion driver file, which exposes all features to the user as a simple collection of Python classes and objects.[37] The user can define experiments directly from Python, leaving all low-level hardware details up to the driver interface.

The core of the QICK system is the tProcessor. This block implements a custom processor with the addition of timed instructions to ensure events are executed at proper times. Pulses for the control and readout of qubits are specified with a complex envelope, which modulates a high-speed carrier. This functionality is implemented by the signal generator block. These blocks have internal memories to upload the IQ tables and fast parallel Direct Digital Synthesis (DDS) blocks for digital upconversion. As Sec. III B explains, the main advantage of the fast DDS is that it generates a tone between 0 and 3 GHz. The nth order harmonics of that tone are typically far away from qubit/cavity resonances and can also be easily filtered out rather than calibrated down with an IQ mixer. The tone images from folding around the Nyquist frequency bands are at predictable locations, and so, the user can frequency-plan to make sure those are far from their signal and thus filtered out. The output of each ADC is connected to a readout block, which implements fast digital downconversion and eight time decimation and filtering, followed by averaging and buffering for signal detection. An additional digital output I/O channel carries digital markers, which are routed to physical connectors to allow for external equipment triggering or fast analog switch control.

The QICK system described in this paper has a fixed number of the three main kinds of blocks: tProcessor, signal generator, and readout. These components have been designed to allow flexibility to address evolving demands. This paper describes the initial version of the firmware. As new configurations are available, these versions can be dynamically loaded between experiments, if necessary, using high-level Python commands.

In a typical experiment, the user will load the waveforms into the signal generator blocks from the PS to the PL. Each readout block is configured before launching the experiment. The program for the tProcessor is uploaded before it begins to run. Even when the tProcessor is programmed with a low-level custom defined assembly language, quantum programs are written using high-level Python classes to ease the description of the experiment. The QICK GitHub repository and QICK documentation website provide demonstrations and documentation of these classes.[31,37] This allows users to directly access all of the capabilities of the system in a user-friendly environment. An effort to make QICK a backend compatible with standard quantum programming languages such as Qiskit is under way.[38]

A. Timed-processor

Figure 8 shows the block diagram of the 64-bit timed-processor (tProcessor). This block is a custom processor with the addition of timed instructions. The tProcessor implements an assembly language grammar including looping, conditional branching, register access, signal generation, and readback. The tProcessor is programmed and controlled with Python APIs.

The tProcessor uses a single master clock, which is implemented using a 48-bit counter. This means that the user can run a single tProcessor program for up to eight days continuously for

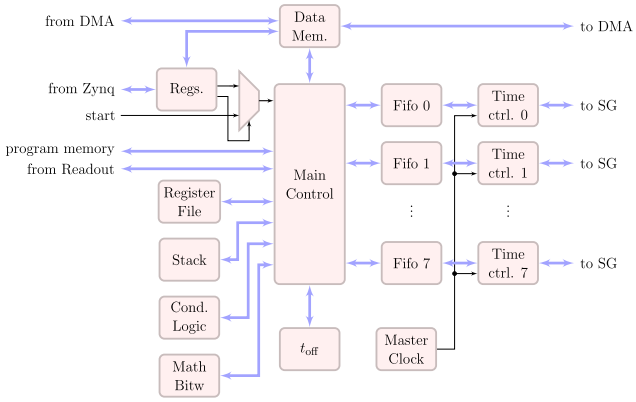

FIG. 8. The microsequencer (tProcessor) firmware block diagram. The tProcessor has a master clock and a time offset register t_{off} that is used to align user instructions in time. Timed instructions are dispatched to the signal generator (SG) (Fig. 9) and readout blocks (Fig. 10). In addition, the user can execute common operations such as register read/write, addition and subtraction, bitwise number manipulation, loop, and memory access. The tProcessor block has a dedicated data memory, which is accessible from the PS using a single AXI read/write or fast DMA transfers. This memory is accessible from the tProcessor itself and can be used as data exchange or for parameters of the experiment.

the implementation presented here. This number could be further increased to allow for longer experiments.

When an instruction with a time tag is found in the program memory, this instruction is dispatched into the corresponding queue of the specified channel (the processor integrates eight output channels). The processor decoding and execution is decoupled from the queue as it can look ahead of time to accelerate the software flow. Instructions dispatched to the timed instruction control queue are executed at the time instructed by the time tag. When the time arrives, the instruction is executed by the timed instruction control sub-block. If the queue of a particular channel gets full, the processor will wait until space becomes available before going further in the program. In other words, non-timed instructions are executed as in a standard processor and timed instructions are dispatched to a queue, which is handled by the time control logic. A more detailed description of the timelines relevant to the QICK architecture is given in Appendix C.

To lower the number of bits necessary to encode the time of a certain instruction, the tProcessor has a special register that indicates the time offset of the time axis. This time offset register is manipulated with specific instructions. When the processor starts, the time offset is set to zero. If a timed instruction is found, it is dispatched with its time tag as the absolute time. As the experiment advances, the time offset register is updated by specific instructions. For example, suppose that the time offset is now 100 and that the same timed instruction is found. Then, the absolute time of the instruction will be 100 plus its time tag. This is dispatched into the queue. This simple structure allows the user to describe times in a very simple manner. Output channels are parallelized, so two or more channels could execute distinct instructions at the same time. Multiple tProcessors could be added to the firmware in the future to control more qubits as they are designed for an external, synchronized start.

B. Signal generator

Figure 9 shows the signal generator FPGA block diagram. The signal generator plays pulse envelopes from a library of pulses created by the user. This block works as an always-ready slave, meaning that if no pulses are queued to be played, its output is chosen to either be zeroed or kept constant from the last played sample. Pulses are played at the time specified by the tProcessor, which guarantees phase coherence. If the specified time is 0, pulses are played immediately. If the queue has been empty and a new pulse is pushed in, there

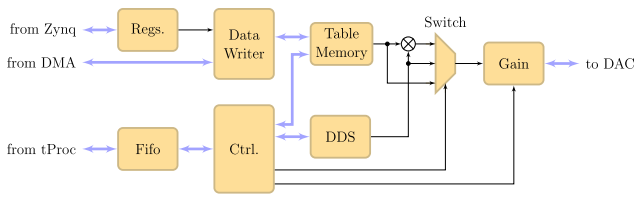

FIG. 9. The signal generator firmware block diagram. The user sends waveform parameters via Python (running on the Zynq PS) to the signal generator block. These waveforms are played at the time specified by the tProcessor block (Fig. 8) to ensure phase coherence. The table memory block stores the I and Q values for the waveform envelope. The fast DDS block synthesizes the tone used for digital upconversion. The switch determines which mode of upconversion to do. The signal then passes through a gain block before entering the DAC block.

is a minimum latency of 20 clocks. The tProcessor uses a standard AXI stream interface to ease system connectivity.

The generator has two main blocks: table memory and DDS. The table memory, which holds I and Q values for the pulse envelope, uses the FPGA's internal Bused Random Access Memory (BRAM) memories. To meet timing for the high-speed DAC, data are interleaved 16-bit words for I and Q. Because the maximum FPGA clock speed is 16 times slower than the maximum DAC speed, the DDS block is highly parallelized to enable the generation of waveforms at the maximum DAC speed. Memory output samples and DDS outputs are complex-multiplied to implement digital upconversion. A single instruction fully specifies the waveform in terms of the start address of the envelope, DDS frequency and phase, pulse duration, output selection, and gain. This allows the output to, for example, switch efficiently from a Gaussian pulse of 20 ns duration and carrier frequency 2 GHz to a pulse of 1 μs and no high frequency modulation and then back to a square envelope pulse at carrier frequency 2.8 GHz. The resolution for the fast DDS is 32-bit, which gives ~1.5 Hz of resolution with a sampling frequency of 6 GHz on the output DAC. It is worth noting that even when the DAC transfers are done at the FPGA clock rate of 384 MHz and 16 samples at a time, the envelope samples can encode up to the full Nyquist bandwidth of the DAC, which in IQ represents 6 GHz.

An important feature of the signal generator block is phase coherence. For proper control of qubits, each signal generator associated with a DAC controls the output pulse phase as follows: an output pulse is sent for implementing a particular rotation of the qubit at a frequency ω_0 and then a second pulse is sent with a different frequency ω_1. If the experiment needs to send another pulse at frequency ω_0, the phase of that waveform needs to be coherent with the previous ω_0 pulse. An easy way of visualizing this property is thinking of a continuous set of sine wave generators that never stop. Then, a phase for a particular frequency at any time corresponds to the phase of the continuously running sine wave of that frequency. In reality, at the time a pulse is output, the signal generator block computes the required phase for a particular frequency with respect to the continuously running sine wave synchronized to master clock origin of time. That allows us to have frames with different frequencies on the same signal generator.

The `mode` of the signal generator can be one-shot or periodic. If `mode = 0`, the signal generator will create a waveform with the specified number of samples one time. If `mode = 1`, the signal generator will keep repeating the actual waveform until the next waveform is read from the queue. The value of `stdsel` determines behavior after all samples are generated. If `stdsel = 0`, the last value of the pulse repeats, and if `stdsel = 1`, the output is set to 0 once all waveforms are completed.

The output of the signal generator is controlled by the switch using the `outsel` parameter. The user can set `outsel = 0` outputs the complex mixing of the DDS and table memory; `outsel = 1` outputs the DDS; `outsel = 2` outputs the table memory; `outsel = 3` outputs 0. This result is multiplied by a gain and sent to the DAC.

C. Readout

The readout block is shown in Fig. 10. The input samples are provided by the parallel digital interface of the high-speed ADC.

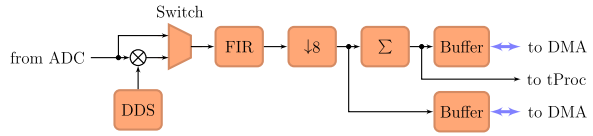

FIG. 10. The readout firmware block diagram. Incoming samples come at high-speed from the ADC block to the readout block. Digital downconversion is performed by multiplying the high-speed incoming samples with a fast DDS. The switch determines which mode of downconversion to do. The resulting signal is low-pass filtered and decimated by 8. The signal is then captured and accumulated in buffers, a process that is controlled by the tProcessor block (Fig. 8) readout trigger.

This block provides eight samples in parallel to allow operating at FPGA speeds.

Similar to the signal generator block, digital downconversion is performed by multiplying the high-speed incoming samples with a fast DDS structure. This parallel DDS has eight individual DDS blocks so that the entire ADC bandwidth is covered in the downconversion process. A switch block allows the user to bypass the downconversion and read the raw data. Upon downconversion, the signal is low-pass filtered and decimated. Filtering and decimation may be tailored to a particular experiment. The filtered signal is fed into the average and raw-buffer blocks. The raw-buffer is mostly used for debugging as it captures the raw, decimated samples. Instead, the average block captures an I and Q pair per readout. The average and capture process is started using a trigger controlled by the tProcessor block. Subsequent triggers will repeat the same average and buffer operation. The averaged values are stored in consecutive memory locations of a circular buffer memory. Data from this buffer is read by the personal computer (PC) using fast DMA transfers. Readout triggers can be executed inside a tProcessor loop, allowing for repeated experiments and sweeping of parameters without software intervention.

The readout time is determined by the user in Python, and the readout is triggered by the tProcessor. The readout pulse offset and length is also configured by the user in Python.

The readout block can implement quantum feedback protocols. An extra AXI stream output is added, which is updated immediately after the average operation. This output is connected to one of the input ports of the tProcessor. The tProcessor can read IQ data from this external port, which can then be used in a conditional branch statement.

IV. PERFORMANCE CHARACTERIZATION

Here, we discuss the RF performance of the signal generator output. Figure 5 shows the schematic of mixer upconversion from IF to RF using the QICK RF board. Since the digitally generated IF can cover 3 GHz of bandwidth in each of the first and second Nyquist zones, we recommend parking the LO at a frequency between 7.5 and 8.5 GHz. The RF board has a low-pass filter that suppresses frequencies over 8 GHz by more than 30 dB of attenuation. We use the lower sideband of the mixer for signal placement. With the LO at 7.5–8.5 GHz, the user can place pulses in a 3 GHz band without

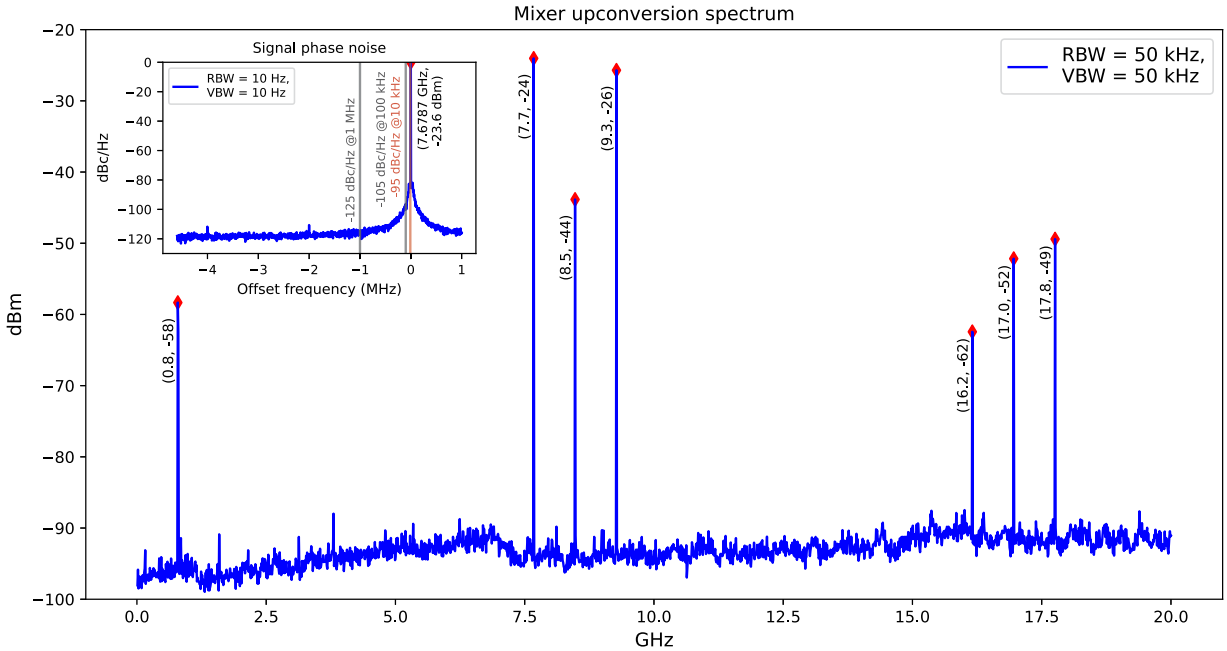

FIG. 11. QICK RF board performance: spectrum of mixer upconversion from IF to RF; inset is signal phase noise. In this measurement, the RF board's additional >8 GHz low-pass filter was disabled so as to display the full upconversion spectrum. Here, an 0.8 GHz IF has been upconverted with an on-board LO set to 8.478 GHz. The mixer spectrum is clean and shows the expected sideband spurs at IF, LO ± IF, and 2 × LO ± IF. We measure −95 dBc/Hz of phase noise at 10 kHz offset from the carrier. Table I summarizes the phase noise measurements.

generating harmonics and having enough LO suppression. Figure 11 shows a demonstration of the clean mixer upconversion spectrum (with the RF board's additional >8 GHz low-pass filter removed). Here, an 0.8 GHz IF has been upconverted with an on-board LO set to 8.478 GHz. From 0.5 MHz to 8 GHz, the mixer spurs are smaller than −60 dBc. Using the LO at lower frequencies is totally viable, but the LO feedthrough will be seen at a power of ∼ −50 dBm. An external low-pass filter or band-pass filter can be added to eliminate the LO feedthrough.

The QICK RF board has been designed to avoid the generation of undesired spurs in the working band of interest and to avoid lengthy calibrations that drift during the course of a single-qubit experiment. One of the main reasons why RF electronics require calibration is that analog IQ mixers have unequal complex gains. Analog IQ mixers are made of two mixers connected by a 90° phase rotation. The amplitude and gain of the I and Q mixers over a large bandwidth (e.g., 4–8 GHz) typically differ by ± half a dB and few degrees. Calibrating the IQ mixer requires multiplying the signal by a frequency-dependent amplitude/phase 2 × 2 matrix. The equations to calculate the matrix values are more involved for multiple tones. For instance, the mixer calibration process for MKID tones achieves sideband rejection of at best 30 dB, and frequent recalibrations are required due to temperature drift. The QICK RF board avoids analog IQ mixers and their calibration requirements by instead using the high frequency DDS and digital IQ mixers in the FPGA, which are inherently balanced. There is no gain error other than a round-off bit that introduces less than −100 dBm of error. Therefore, unlike most qubit controllers, the QICK does not require any mixer calibration.

The analog RF mixer on the QICK RF board is a non-IQ DSM (double sideband mixer). The DSM generates the two sidebands at the frequencies LO ± IF. The mixer has some direct IF and LO leakage into the RF output, and the mixer nonlinearity generates spurs at the frequencies nLO ± m IF for integers n and m. Figure 5 shows how the QICK RF board filters all undesired outputs. As mentioned in Sec. III B, the fast DDS allows the user to place an IF pulse anywhere in a 3 GHz wide spectrum (the left rectangle in Fig. 5 before the mixer symbol). For instance, as in Fig. 5, the ±3 GHz IF spectrum could be shifted up by an 8.5 GHz on-board LO and the two sidebands would be at 5–8 and 9–12 GHz. The IF and LO feedthrough and nLO ± m IF products do not fall inside the 5–8 GHz band and are easily filtered. We have 500 MHz of room left between the LO and the LSB of interest, which allows for 35+ dB of filtering. Moving the LO in the 7.5–8.5 GHz band enables us to place our upconverted signal in the 4–8 GHz band. Figure 11 shows the unfiltered mixer output using a 8.478 GHz LO and 800 MHz IF. A typical 17 dB LO is attenuated by 66 dB (to −48 dBm). When the LO is placed at 8.478 GHz, the Mini-Circuits LFCW-6000+ adds another 45 dB of attenuation (to −93 dBm). The noise floor in the 4–8 GHz band is −135 dBc/Hz. The phase noise near the carrier is shown in the inset of Fig. 11. Table I shows phase noise measurements from 100 Hz to 10 MHz.

The QICK's measured phase noise of −95 dBc/Hz at 7.6787 GHz carrier at 10 kHz offset is comparable to that of commercial qubit controllers, which use direct digital synthesis (e.g., the Zurich Instruments SHFQA has a measured phase noise of −96 dBc/Hz at 8 GHz carrier at 10 kHz offset[39]). For the highly dynamic pulse generation necessary for most qubit control applications, the QICK measured phase noise is sufficient for low-gate-error operations. For instance, it is below the value specified in Ref. 40 for >99.9% fidelity single-qubit control (−116 dBc/Hz at a 1 MHz offset from the carrier). In addition, one can compute its corresponding qubit dephasing spectral density using the method in Ref. 41 and find it to generally be several orders of magnitude lower than that of the lab-grade local oscillator studied in that paper. For applications that require ultra-low phase noise, the user can substitute the QICK RF board with their choice of upconversion circuit whose dedicated local oscillator has lower phase noise. The QICK phase noise is limited by the RF mixer since we measured the phase noise directly out of the RFSoC DAC to be lower by 15–20 dB. In the future RFSoC generations, faster DACs, which entirely eliminate the use of IQ mixers, may help to reduce phase noise.

TABLE I. QICK RF board phase noise measurements. In these measurements, the carrier tone frequency was 7.6787 GHz.

Delta freq. (Hz)	Phase noise (dBc/Hz)
100	−70
1k	−80
10k	−95
100k	−105
1M	−125
10M	−125

We also measured gain stability of the QICK vs time for a total of 16 h. In this test, the controller played a single tone in loopback mode and the gain was measured in ADC digital units. No calibration was performed during this time besides the default background gain and temperature calibration that occurs in the RFSoC ADC.[42] The board was sitting at room temperature with no additional temperature control. Over that time period, the QICK measured gain had an RMS value of 0.07%, which is sufficient for low-gate-error operations. For instance, this analog gain stability surpasses the value of 0.22% specified in Ref. 40 for >99.9% fidelity single-qubit control.

A. Latency

Low latency is important for running deep circuits on noisy qubits that decohere quickly. If the qubit experiment is conditional upon the previous value read out from the system, the readout latency must also be included. Table II details measured latency numbers for the QICK. As mentioned in Sec. I, the RFSoC DAC and ADC modules have a cascade of processing blocks that can be used or bypassed. The latency depends on the number of blocks being used. Figure 12 shows a simplified block diagram of the DAC. The ADC is very similar, but the interpolation is replaced by a decimation. The round trip latency including the DAC, ADC, and the digital AXI interfaces associated with the DAC and ADC was measured using an ILA (Internal Logic Analyzer) running at 512 MHz. When we bypass the interpolation/decimation and filters, the latency is 90 ns (46 ILA clocks). Half of that latency is due to the DAC, and half is due to the ADC. When the upconversion and down-conversion are enabled, the round trip latency increases to 113 ns (58 ILA clocks). When the ×8 interpolation/decimation and filters

TABLE II. Latency measurements taken to help develop the QICK firmware. Note that neither the NCO or the ×8 interpolation/decimation and filters are enabled in the current version of the QICK firmware although they may be used in the future firmware versions.

Sampling freq. (MHz)	Functions	Latency (ns)
ADC:4096 DAC:6144	ADC and DAC with all digital features bypassed	90
ADC:4096 DAC:6144	ADC and DAC with NCO enabled	113
ADC:4096 DAC:6144	ADC and DAC, NCO enabled, and interpolation/decimation enabled	117
ADC:3072 DAC:6144	Conditional evaluation and address jump	42
ADC:3072 DAC:6144	Next pulse latency if address jump = TRUE	52

are included, the latency goes up to 117 ns (60 ILA clocks) round trip. The loopback between DAC and ADC was made with a short coax cable that was a few inches long.

Another test was created to measure the latency of the logic in feedback readout mode. A readout pulse was output and fedback into an ADC while a second DAC and a marker were used to measure times. The experiment shows that the conditional evaluation and address jump latency is 16 clock cycles. At the current firmware ADC sample rate, that totals 42 ns. If an address jump occurs, the next pulse latency is 20 clock ticks, which equals 52 ns. Summing the measurements, the total latency of the QICK is 184–211 ns depending on the DAC and ADC configuration. A new version of the tProcessor is under development using pipelining to reduce the logic latency due to internal processing and instruction branching.

B. Characterization of a transmon qubit

Describing superconducting qubit dynamics largely exceeds the scope of this paper. There are hundreds of excellent papers on the subject such as Refs. 43–52. In this section, we will summarize only the information necessary to explain the functionality of the QICK.

The classical control and readout of superconducting qubits allows for information to travel from the classical world to the quantum world and vice versa. High-quality classical control enables both (a) the preparation of large coherent quantum states composed of entangled qubits and (b) the manipulation and readout of these large quantum states. Such tools are integral to the execution of algorithms such as error correction and factoring as well as to the preparation and measurement of states for quantum simulation.

Superconducting qubit systems are susceptible to noise-induced decoherence, and faulty qubit control (for instance, driving the qubit with spurious frequency components) can lead to driving unwanted transitions that cause the system to decohere. Noisy qubit-readout is also a source of faulty control. Typically, the readout noise is determined by the first amplifier in the readout chain [a cryogenic Low Noise Amplifier (LNA) such as a HEMT] and not by the warm electronics. Therefore, the QICK RF board was designed to ensure that the input noise of the warm electronics would be less than the cryogenic LNA output noise to avoid lowering the signal to noise ratio (SNR) of the readout signal. In addition, the QICK RF board's efficient filtering of the warm readout electronics was designed to enhance the readout signal's SNR.

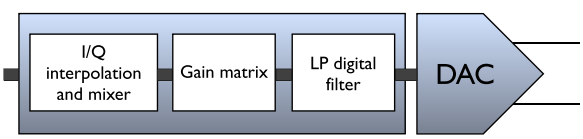

FIG. 12. Simplified RFSoC DAC module block diagram. The ADC block diagram is similar, but the interpolator block is replaced by a decimator block.

Superconducting qubits are typically coupled to a resonant RF cavity. The qubit state is read out using a quantum non-demolition (QND) technique in which the state of the qubit is projected onto the RF cavity.[53] Repeated QND measurements of the qubit yield the same result. The readout electronics is broadband (i.e., several GHz); therefore, several qubits, typically 100 MHz bandwidth, can be frequency multiplexed and read out simultaneously. A typical readout scheme measures transmitted power S_{21} through the cavity that is coupled to the qubits.

The high output bandwidth (4 GHz) of the QICK RF DACs and RF ADCs (1) improves the overall quality of qubit control by enabling the direct synthesis of spectrally pure qubit control pulses in the second Nyquist zone and (2) allows the controller to multiplex qubit control and readout. The multiplexing functionality is possible with the current QICK system but has not yet been implemented in an experiment. Our eventual aim is to multiplex up to 20 qubits (with 100 MHz bandwidth and 200 MHz separation) on the same control line, increasing the number of qubits to 100+ per board and several thousand in a modest system with few tens of boards.

The digital part of the QICK (loaded onto the ZCU111 evaluation board with the standard Xilinx XM500 RFMC balun card) was deployed and used to control qubits in the Schuster lab at the University of Chicago's James Franck Institute and Pritzker School of Molecular Engineering. The controller took high-quality single-qubit data even without the custom RF board, which was still being developed at Fermilab at the time the measurements were taken. The custom RF board was replaced by connectorized amplifiers, attenuators, mixers, and filters, which did the necessary amplification and up/downconversion. The overall controller performance was found to be on par with commercial qubit controllers that are ten times as expensive. The QICK controller was also straightforward for researchers to use particularly due to its ability to directly synthesize carrier frequencies of up to 6 GHz. Compared to the conventional

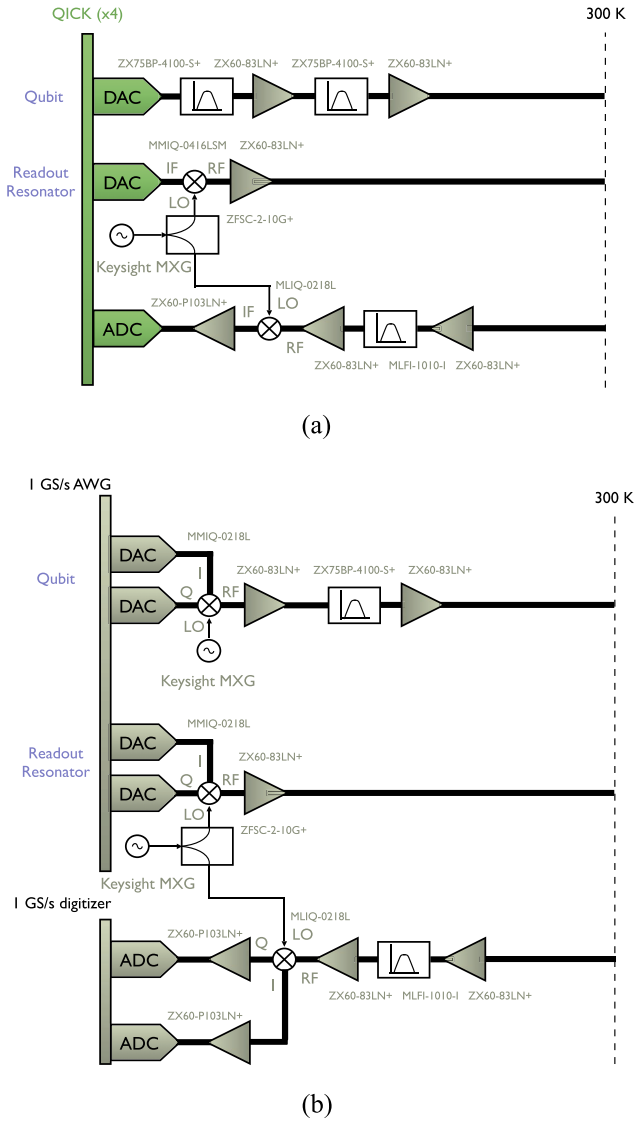

FIG. 13. Comparison of wiring diagrams with the QICK ZCU111 evaluation board (a) and with a conventional AWG setup (b). The QICK has eight RF DACs, which can be configured in multiple ways. Using the QICK, two RF DACs are needed to control the qubit and its readout resonator, respectively. In addition, the qubit control pulses (<6 GHz) can be directly synthesized without the use of an analog mixer. This setup was used to gather the data shown in Fig. 14. Note that the QICK RF board was not used because it was still under development. In a conventional AWG setup, four DACs would be needed to control a qubit and its readout resonator, and both the qubit and the readout resonator pulses would need to be upconverted with analog mixers.

qubit controller used in the Schuster lab, the QICK required half the DAC channels and half the amount of analog upconversion (Fig. 13). The room-temperature and cryogenic setup originally used to measure this qubit before the QICK was introduced is shown in Fig. S2 of Ref. 32.

The QICK was used to control and read out a 3D transmon qubit that is being used as a dark matter detector.[32] The readout resonator pulses were generated and upconverted using a mixer with an IF frequency of 100 MHz. The IF frequency was swept to perform a resonator spectroscopy as shown in Fig. 14(a). The qubit pulses were directly synthesized operating one of the DAC channels in the second Nyquist zone. The qubit probe frequency was swept around 4.743 GHz followed by a readout tone to obtain Fig. 14(b). The amplitude or power Rabi measurement was performed by varying the pulse amplitude of a 100 ns (σ = 25 ns) long Gaussian pulse [Fig. 14(c)]. Ramsey interferometry was performed by preparing the qubit in a superposition state with a $\frac{\pi}{2}$ pulse followed by a variable delay (τ) before applying another $\frac{\pi}{2}$ pulse with a phase advanced by $\Delta\phi = \omega \times \tau$, where ω is the Ramsey frequency [Fig. 14(d)]. A T_1 measurement was performed by preparing the qubit in its excited state with a π pulse followed by a variable delay (τ) before measuring the qubit's state [Fig. 14(d)]. The single shot IQ readout values were acquired for the qubit prepared in ground (blue dots) and excited state (red dots) to estimate the readout fidelity. The resulting distribution was fitted with a bi-modal Gaussian function to extract the single shot fidelity, F = 94.7% without any parametric amplification as shown in Fig. 14(e). Finally, we characterize the fidelity of our single-qubit gates through a randomized benchmarking protocol. The gates were randomly chosen from this gate set {I, X, Y, Z, X/2, −X/2, Y/2, −Y/2, Z/2, −Z/2} where Z and ±Z/2 gates were implemented virtually by advancing the phase of subsequent gates. For each sequence length, we use 30 randomized sequences, each containing a recovery gate to bring the qubit back to its ground state before performing a readout. Each individual data point is averaged over 3000 shots. The average gate fidelity was found to be $\mathcal{F}_{avg}$ = 99.93% ± 0.01% as shown in Fig. 14(f). The coherence-limited gate fidelity is estimated to be $\mathcal{F}_{\text{lim}}$ = 99.96%. The duration and integration time of the readout pulse for these measurements were 3 μs. This is related to the qubit-readout dispersive shift ($\frac{\chi}{2\pi} \sim 350$ kHz), which is limited by the device design and not by the control hardware. Interested readers may follow excellent review articles describing standard qubit characterization experiments.[54,55]

Controlling this transmon with the QICK was not found to degrade its coherence relative to the coherences measured with two different commercial systems: a conventional Keysight arrayed waveguide grating Arbitrary Waveform Generator (AWG)-based system and the Quantum Machines OPerator X model. The T_1 decay time measured with the QICK was 119.5 ± 1.6 μs (versus 108 ± 18 μs measured with the AWG-based system[32] and 122.6 ± 2.5 μs with the OPX). The T_2 decay time measured with the QICK was 148.6 ± 4.6 μs (versus 61 ± 4 μs measured with the AWG-based system and 155.9 ± 3.3 μs with the OPX). The AWG-based system measured a significantly lower T_2 because there was increased readout population for that cooldown, which occurred prior to the cooldown where the QICK and the OPX were each used to measure the device. The remaining discrepancies between these values can be attributed to the well-documented phenomenon of superconducting qubit coherence fluctuating over long time scales.[56,57] Additionally, the readout fidelity measured with the QICK was 94.7%, which is comparable to the two other measurements of readout fidelity (95.8% ± 0.4% measured with the AWG-based system and 95% measured with the OPX). The remaining discrepancies between these values can be attributed to differences in the readout pulse length and envelope

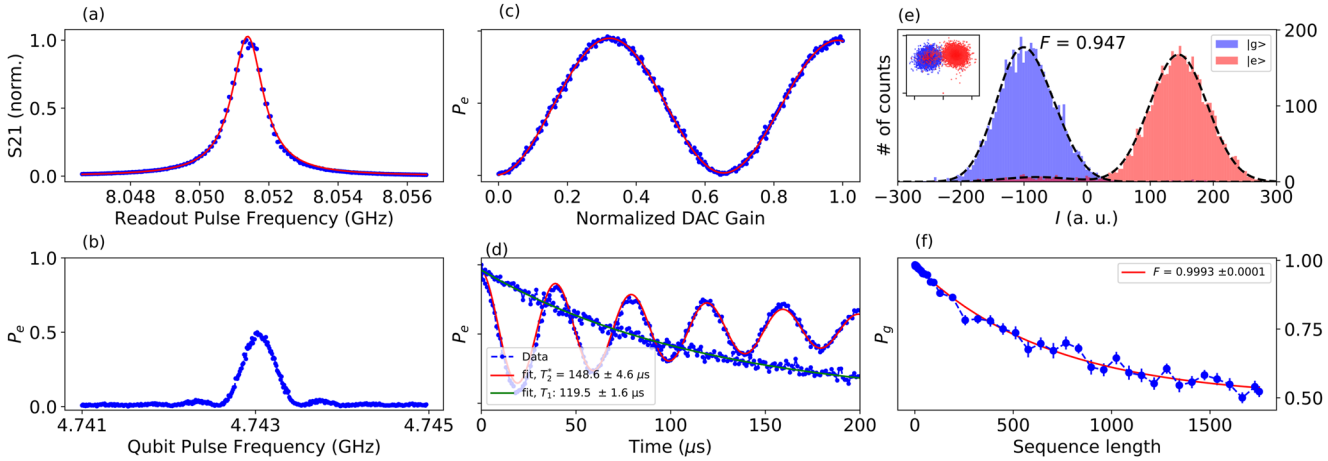

FIG. 14. The QICK was used to characterize a transmon qubit dispersively coupled to a readout cavity. Figures (a) and (b) show readout cavity and qubit spectroscopy measurements, and figure (c) shows qubit Rabi oscillations. Figure (d) show qubit T_1 and T_2 measurements (of 119 and 148 μs, respectively). Figure (e) (inset) shows single shot digitized values when the qubit is prepared in ground and excited states. Histograms were fitted with a bi-modal Gaussian distribution function, resulting in 94.7% fidelity with no parametric amplification. Figure (f) shows a randomized benchmarking protocol averaged over 30 unique random sequences with 3000 shots per data point. The average gate fidelity is $\mathcal{F}_{avg}$ = 99.93% ± 0.01%, which approaches the estimated coherence-limited gate fidelity of $\mathcal{F}_{lim}$ = 99.96%. All measurements were performed entirely on the QICK by running a Jupyter notebook on-board, which compiled experiments and acquired data.

shape as well as differences in digitization methods used for the respective experiments conducted.

The QICK has also been deployed in the Schuster lab to benchmark heavy fluxonium qubits with fast flux pulses instead of standard RF microwave pulses. The lab recently demonstrated this technique in Ref. 35 with a conventional control setup and was afterward able to reproduce their results with the QICK. Such experiments require the QICK DACs to be in DC coupled mode so that they can generate fast unmodulated pulses.

V. SUMMARY AND FUTURE WORK

We introduced an RFSoC-based qubit controller called the Quantum Instrumentation Control Kit (QICK) that is capable of directly synthesizing qubit control pulses with IF frequencies up to 3 GHz (6 GHz) in the first (second) Nyquist zone. The QICK can be used to control multi-qubit systems. The low cost of the controller, roughly $15 000 USD for eight RF DAC channels and eight RF ADC channels, makes it useful for scaling up qubit experiments in academic laboratories.

Despite the short development time (1.5 years), our team has added low and high-level functionalities to operate complex qubit systems and experiments. The QICK provides an integrated solution for qubit biasing, control, and readout. All RF and synchronization devices are included on-board. Users can freely access the QICK Github repository with controller firmware, software, and documentation.[31]

Our future work focuses on hardware, firmware, and software improvement and improved functionality. On the hardware side, we are exploring the use of the Xilinx ZCU216 board hosting a ZU49DR RFSoC Gen3 FPGA.[58] The ZU49DR DACs run at 10 GS/s with increased analog bandwidth, >6 GHz. Many more typical qubit experiments could now be run without the need of an external analog mixer, further simplifying the setup. The ZU49DR has 16 DACs and 16 ADCs, which implies scaled-up firmware functionality and throughput. To allow for faster qubit experiments with more channels, we will increase the number of tProcessor microsequencers in the FPGA. We will allocate one microsequencer per channel or per every few channels without compromising FPGA resources. To improve the flexibility of the signal generator, we will (1) integrate the variable length and interpolation envelope engines, which are currently in a beta phase and (2) develop drivers for very long AWG pulses requiring DDR4 and interpolation. We will also develop and implement optimal filters for readout.

Much of our future work focuses on the software and software–firmware interface. We will integrate the QICK with a simulator and debugger to cut the qubit experiment program development time. We will also integrate our current software with high-level software packages such as Qiskit and OpenQASM,[38,59] which will help developers.

ACKNOWLEDGMENTS

This article has been authored by Fermi Research Alliance, LLC, under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics, with support from its QuantISED program and from the National Quantum Information Science Research Centers, Quantum Science Center. This work was funded, in part, by EPiQC, an NSF Expedition in Computing, under Grant No. CCF-1730449. This work was supported by the Army Research Office under Grant No. W911NF1910016. S. Sussman was supported by the Department of

Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program. A. Agrawal was supported by the Heising-Simons Foundation. Salvatore Montella was supported by the DOE SQMS and from the Quantum Science Center and Superconducting Quantum Materials and Systems Center. The Fermilab team thanks Gaston Gutierrez (Fermilab) for his help in bridging the gap between engineering and quantum mechanics. The authors acknowledge the National Quantum Information Science Research Centers Q-NEXT and SQMS under Contract No. DE-AC02-07CH11359 and thank C2QA members who participated in discussions.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: TWO-BOARD SYNCHRONIZATION

Two ZCU111 boards were synchronized to an external rubidium stable source. On each board, a DAC output was used to generate a continuous single 10 MHz tone using a fast DDS and a constant envelope. The output of a DAC from one ZCU111 was used to drive the horizontal input of a Tektronix oscilloscope, and a DAC output from the other ZCU111 was used to drive the vertical scope input. Figure 15 shows the typical Lissajous figure. The plot remains stable independently of time (e.g., hours). Turning the synchronization between the two boards off and on again only modifies the initial phase of the Lissajous figure, which remains stable.

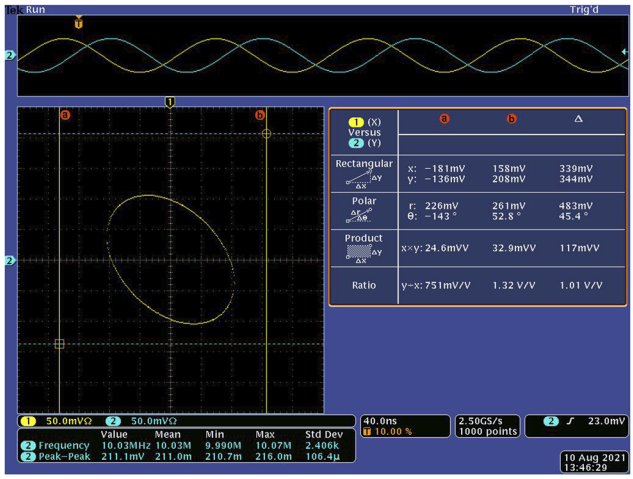

FIG. 15. Lissajous figure from two-board synchronization. The output of a DAC from one ZCU111 was used to drive the horizontal input of a Tektronix oscilloscope, and a DAC output from the other ZCU111 was used to drive the vertical scope input.

APPENDIX B: BANDWIDTH TABLE

Bandwidths of different aspects of the QICK system (Table III).

APPENDIX C: THE QICK ARCHITECTURE TIMELINES

There are two different timelines in the QICK architecture: the tProcessor timeline of decision-making about the placement of events according to the master clock and the master clock timeline itself. These timelines are decoupled although not totally independent since the tProcessor has to be ahead of the master clock timeline to avoid the master clock timeline being under-run and waiting for instructions that arrive very late. The tProcessor timeline is managed as follows:

i. Compiled quantum algorithm instructions are written into the tProcessor memory.
ii. The tProcessor decodes instructions. If the instruction can be executed immediately (e.g., queue a pulse generator request and configure hardware or firmware parameters), it does so.
iii. The tProcessor waits for specific events such as pulse generator queues being full, channel readout triggers, feedback from a readout, or a forced wait.

The master clock timeline is managed as follows:

i. All signal generator pulses are played at the exact time as specified by the time in the queue.

TABLE III. Bandwidths of different aspects of the QICK system. Note: The first three lines in the table reflect the allowable digital and analog BW of the RFSoC board. The last three lines reflect the analog BW on the QICK RF custom board. The filters on the RF board have been chosen to minimize RF noise of current experiments. Those filters can be replaced to increase or reduce the analog BW as needed.

Device	Frequency (GHz)	Comments
Digital DDS	0–3, 3–6	First, second Nyquist zones
DAC NCO	0 to F_s	Not used in current version of firmware
DAC analog BW	6	Measured using broadband white Gaussian noise
Analog mixer LO frequency	2–15	Tunable range
RF low-pass filter	1.8	LFCN-1800+ fixed but could be replaced
RF high-pass filter	6.3	LFCW-6300+ fixed but could be replaced

ii. Every signal generator has its own queue, and they are all synchronized to the same master clock. More than one signal generator may output a pulse at the same time or overlapping time with other channels.
iii. Readout channels, upon a tProcessor trigger, are synchronized to the master clock.

REFERENCES

[1] P. W. Shor, SIAM J. Comput. **26**, 1484–1509 (1997).
[2] C. Gidney and M. Ekerå, Quantum **5**, 433 (2021); arXiv:1905.09749.
[3] C. H. Bennett and G. Brassard, Theor. Comput. Sci. **560**, 7–11 (2014).
[4] J. Yin, Y. H. Li, S. K. Liao, M. Yang, Y. Cao, L. Zhang, J. G. Ren, W. Q. Cai, W. Y. Liu, S. L. Li, R. Shu, Y. M. Huang, L. Deng, L. Li, Q. Zhang, N. L. Liu, Y. A. Chen, C. Y. Lu, X. B. Wang, F. Xu, J. Y. Wang, C. Z. Peng, A. K. Ekert, and J. W. Pan, Nature **582**, 501 (2020).
[5] C. S. Wang, J. C. Curtis, B. J. Lester, Y. Zhang, Y. Y. Gao, J. Freeze, V. S. Batista, P. H. Vaccaro, I. L. Chuang, L. Frunzio, L. Jiang, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. X **10**, 021060 (2020).
[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195–202 (2017).
[7] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina, and C. Monroe, "Fault-tolerant operation of a quantum error-correction code," arXiv:2009.11482 [quant-ph] (2021).
[8] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Annu. Rev. Condens. Matter Phys. **11**, 369–395 (2020).
[9] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi *et al.*, Nat. Nanotechnol. **13**, 102–106 (2017).
[10] R. Schirhagl, K. Chang, M. Loretz, and C. L. Degen, Annu. Rev. Phys. Chem. **65**, 83 (2014).
[11] I. Bloch, Nat. Phys. **1**, 23 (2005).
[12] L. M. Sager, S. E. Smart, and D. A. Mazziotti, Phys. Rev. Res. **2**, 043205 (2020).
[13] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. R. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. N. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandra, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, Nature **574**, 505 (2019).
[14] S. Asaad, C. Dickel, N. K. Langford, S. Poletto, A. Bruno, M. A. Rol, D. Deurloo, and L. DiCarlo, npj Quantum Inf. **2**, 16029 (2016).
[15] T. Walter, P. Kurpiers, S. Gasparinetti, P. Magnard, A. Potočnik, Y. Salathé, M. Pechal, M. Mondal, M. Oppliger, C. Eichler *et al.*, Phys. Rev. Appl. **7**, 054020 (2017).
[16] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. M. Girvin, L. Jiang, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, Nature **536**, 441 (2016).
[17] ARTIQ Website, https://m-labs.hk/experiment-control/artiq, 2021.
[18] C. A. Ryan, B. R. Johnson, D. Ristè, B. Donovan, and T. A. Ohki, Rev. Sci. Instrum. **88**, 104703 (2017).
[19] Keysight Quantum Solutions, https://www.keysight.com/us/en/solutions/emerging-technologies/quantum-solutions.html, 2021.
[20] Zurich Instruments Website, https://www.zhinst.com/americas/en, 2021.
[21] Quantum Machines Website, https://www.quantum-machines.co, 2021.
[22] Y. Xu, G. Huang, J. Balewski, R. Naik, A. Morvan, B. Mitchell, K. Nowrouzi, D. I. Santiago, and I. Siddiqi, in *IEEE Transactions on Quantum Engineering* (IEEE, 2021), Vol. 2, pp. 1–11.
[23] W. D. Kalfus, D. F. Lee, G. J. Ribeill, S. D. Fallek, A. Wagner, B. Donovan, D. Riste, and T. A. Ohki, IEEE Trans. Quantum Eng. **1**, 1 (2021).
[24] S. W. Jolin, R. Borgani, M. O. Tholén, D. Forchheimer, and D. B. Haviland, Rev. Sci. Instrum. **91**, 124707 (2020).
[25] Xilinx RFSOC Website, https://www.xilinx.com/products/silicon-devices/soc/rfsoc.html, 2021.
[26] R. Gebauer, N. Karcher, and O. Sander, Phys. Rev. Appl. **15**, 064030 (2021).
[27] K. H. Park, Y. S. Yap, Y. P. Tan, C. Hufnagel, L. H. Nguyen, K. H. Lau, S. Efthymiou, S. Carrazza, R. P. Budoyo, and R. Dumke, "ICARUS-Q: A scalable RFSoC-based control system for superconducting quantum computers," arXiv:2112.02933 [quant-ph] (2021).
[28] Intermodulation Products Website, https://intermodulation-products.com/products/microwave-platforms, 2021.
[29] Python productivitye for zynq, https://pynq.io, 2021.
[30] Zynq Ultrascale+ RFSOC Product Data Sheet: Overview (DS889), https://www.xilinx.com/support/documentation/data_sheets/ds889-zynq-usp-rfsoc-overview.pdf, 2020.
[31] QICK Github repository, https://github.com/openquantumhardware/qick, 2021.
[32] A. V. Dixit, S. Chakram, K. He, A. Agrawal, R. K. Naik, D. I. Schuster, and A. Chou, Phys. Rev. Lett. **126**, 141302 (2021).
[33] A. B. Walter, N. Fruitwala, S. Steiger, J. I. Bailey, N. Zobrist, N. Swimmer, I. Lipartito, J. P. Smith, S. R. Meeker, C. Bockstiegel, G. Coiffard, R. Dodkins, P. Szypryt, K. K. Davis, M. Daal, B. Bumble, G. Collura, O. Guyon, J. Lozi, S. Vievard, N. Jovanovic, F. Martinache, T. Currie, and B. A. Mazin, Publ. Astron. Soc. Pac. **132**, 125005 (2020).
[34] N. Fruitwala, P. Strader, G. Cancelo, T. Zmuda, K. Treptow, N. Wilcer, C. Stoughton, A. B. Walter, N. Zobrist, G. Collura, I. Lipartito, J. I. Bailey, and B. A. Mazin, Rev. Sci. Instrum. **91**, 124705 (2020).
[35] H. Zhang, S. Chakram, T. Roy, N. Earnest, Y. Lu, Z. Huang, D. K. Weiss, J. Koch, and D. I. Schuster, Phys. Rev. X **11**, 011010 (2021).
[36] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen, Rev. Mod. Phys. **79**, 1217 (2007).
[37] QICK documentation website, https://qick-docs.readthedocs.io/en/latest, 2021.
[38] IBM Qiskit library, https://qiskit.org, 2021.
[39] Zurich Instruments SHFQA Specifications, https://docs.zhinst.com/shfqa_user_manual/specifications.html, 2021.
[40] J. P. V. Dijk, B. Patra, S. Pellerano, E. Charbon, F. Sebastiano, and M. Babaie, IEEE Trans. Circuits Syst. **67**, 5380 (2020).
[41] H. Ball, W. D. Oliver, and M. J. Biercuk, "The role of master clock stability in quantum information processing," npj Quantum Inf. **2**, 16033 (2016).
[42] Zynq Ultrascale+ RFSOC RF Data Converter Product Guide (PG269), https://www.xilinx.com/support/documentation/ip_documentation/usp_rf_data_converter/v2_6/pg269-rf-data-converter.pdf, 2021.
[43] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. A **76**, 042319 (2007).
[44] J. Gambetta, A. Blais, D. I. Schuster, A. Wallraff, L. Frunzio, J. Majer, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. A **74**, 042318 (2006).
[45] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai, Nature **398**, 786 (1999).
[46] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[47] A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, J. Majer, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. Lett. **95**, 060501 (2005).
[48] J. M. Martinis, Quantum Inf. Process. **8**, 81 (2009).
[49] J. Preskill, in Annual Symposium on Foundations of Computer Science–Proceedings, 1997.
[50] D. P. DiVincenzo, Fortschr. Phys. **48**, 771 (2000).
[51] R. J. Schoelkopf and S. M. Girvin, Nature **451**, 664 (2008).
[52] M. H. Devoret and R. J. Schoelkopf, Science **339**, 1169 (2013).
[53] D. I. Schuster, "Circuit quantum electrodynamics," Ph.D. thesis, Yale University, 2007.
[54] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, Appl. Phys. Rev. **6**, 021318 (2019).

[55]M. Naghiloo, "Introduction to experimental quantum measurement with superconducting qubits," arXiv:1904.09291 [quant-ph] (2019).
[56]A. P. Place, L. V. Rodgers, P. Mundada, B. M. Smitham, M. Fitzpatrick, Z. Leng, A. Premkumar, J. Bryon, A. Vrajitoarea, S. Sussman, G. Cheng, T. Madhavan, H. K. Babla, X. H. Le, Y. Gang, B. Jäck, A. Gyenis, N. Yao, R. J. Cava, N. P. de Leon, and A. A. Houck, Nat. Commun. **12**, 1779 (2021).
[57]M. Carroll, S. Rosenblatt, P. Jurcevic, I. Lauer, and A. Kandala, "Dynamics of superconducting qubit relaxation times," arXiv:2105.15201 [quant-ph] (2021).
[58]Xilinx and Avnet ZCU216 evaluation board user guide, https://www.xilinx.com/support/documentation/boards_and_kits/zcu216/ug1390-zcu216-eval-bd.pdf, 2020.
[59]IBM Qiskit OpenQASM library, https://github.com/Qiskit/openqasm, 2021.