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

Leandro Stefanazzi, Ken Treptow, Neal Wilcer, Chris Stoughton, Salvatore Montella, Collin Bradford, and Gustavo Cancelo
Fermi National Accelerator Laboratory, Batavia IL, United States

Shefali Saxena
Argonne National Laboratory, Lemont IL, United States

Horacio Arnaldi
Instituto Balseiro, CNEA, Argentina

Sara Sussman and Andrew Houck
Department of Physics and Department of Electrical Engineeering, Princeton University, Princeton NJ, United States

Ankur Agrawal, Helin Zhang, Chunyang Ding, and David I Schuster
Department of Physics and Pritzker School of Molecular Engineering, University of Chicago, Chicago IL, United States

(Dated: October 4, 2021)

We introduce a Xilinx 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 (RF System-on-Chip) FPGA [1], 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 Clifford gate fidelity of $\mathscr{F}_{avg} = 99.93\%$. All of the schematics, firmware, and software are open-source [2].

I. INTRODUCTION

Quantum computers are predicted to outperform classical computers in problem domains such as decryption [3, 4], secure communication [5, 6], quantum chemistry [7] and machine learning [8]. Quantum bits (qubits), have been developed in several different platforms including trapped ions [9], superconducting qubits [10], semiconductor quantum dots [11], color centers [12], and neutral atoms [13]. 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 [14, 15], prohibitive for academic labs and small startups. We have developed a RF/FPGA-based system that is flexible enough to support different platforms and is affordable to academic laboratories.

There have been several platforms developed which integrate fast RF DACs and ADCs with Field-Programmable Gate Array (FPGA)s. Early academic efforts in the superconducting qubit community were made at Delft for pulse routing [16] and at ETH Zurich for real-time processing of measurements [17]. The first demonstration of quantum error correction to show improvement over the physical constituents was enabled by an FPGA controlled data acquisition system [18]. The ion trap community has developed the FPGA-based ARTIQ system [19] 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 like BBN [20], Keysight [21], Zurich Instruments [22], and Quantum Machines [23]. An open-source qubit controller was recently developed by LBNL and UC Berkeley [24]. These solutions all use a conventional method by which the FPGA-controlled DAC synthesizes the IF envelope and is upconverted via 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 [25], eliminating the need for meticulous calibration [26].

In this work we take advantage of a newly developed FPGA platform, the RFSoC from Xilinx [27], which integrates high-speed DACs, ADCs, programmable FPGA logic, and a conventional microprocessor in the same package. The Quantum Instrumentation Control Kit (QICK) combines FPGA firmware for real-time processing and a Python interface running the PYNQ [28] operating system. The initial implementation of the QICK uses the ZCU111 [29] evaluation board with 8 DACs and ADCs. This allows pulse synthesis up to 6 GHz directly using the XM500 RFMC balun card provided in the evaluation kit, and up to 15 GHz 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

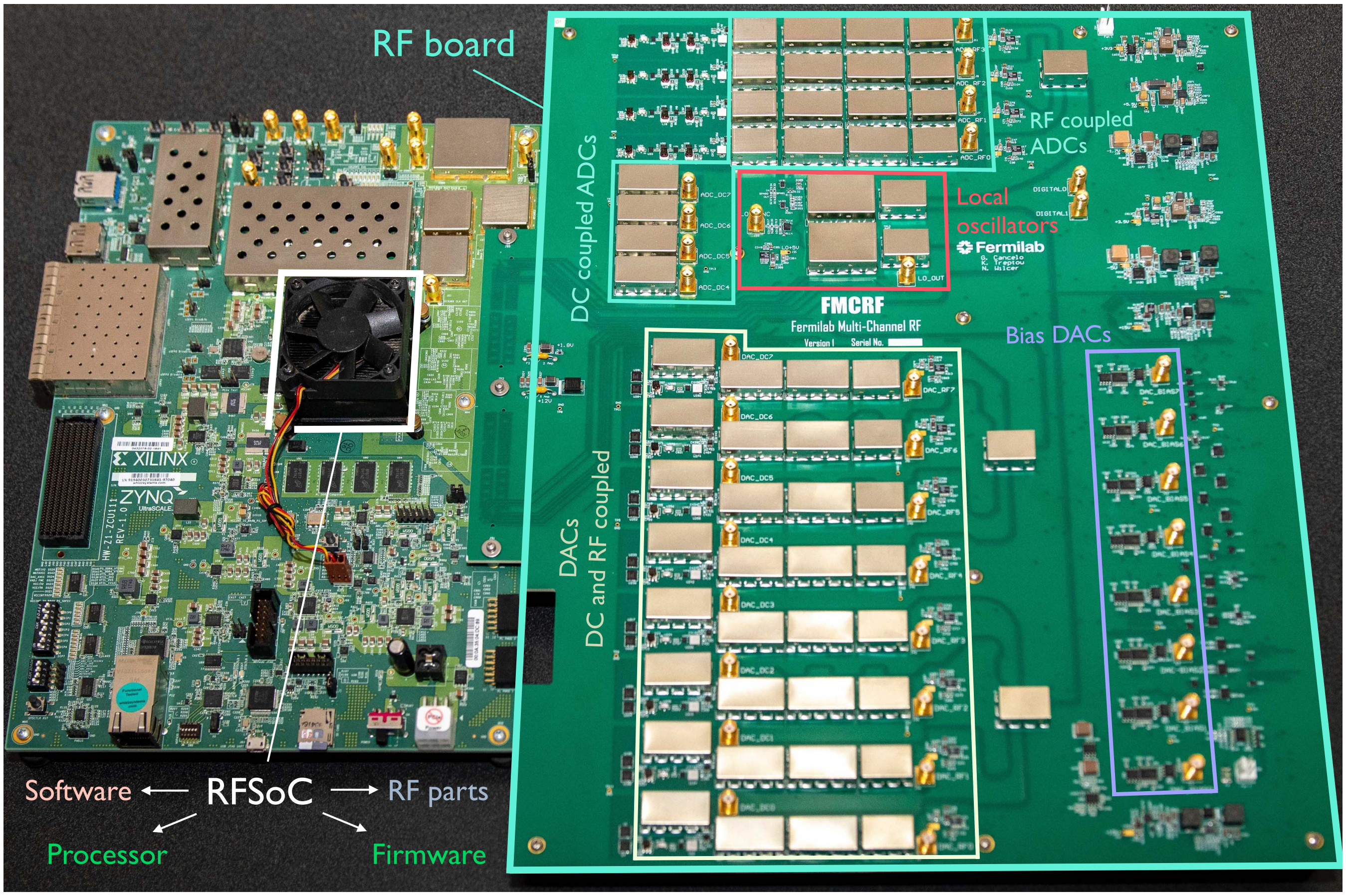


FIG. 1. The Quantum Instrumentation Control Kit (QICK). The QICK consists of two pieces of hardware: the commercial ZCU111 RFSoC evaluation board (left), which connects to the QICK RF board (right) which can be used for additional signal up/downconversion, amplification and filtering.

the input. The system is open-source [2] and can run on commercially available hardware at a price significantly cheaper than commercial offerings at this time.

The rest of the manuscript is organized as follows: in section II we describe the hardware and its capabilities. In section III we describe the controller architecture, and the microsequencer with precision timing blocks, along with the digital upconversion and downconversion chains. Next, in section 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 section 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 [30].

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 of the QICK is the FNAL-Gen2 fMESSI system which generated two chronograph instruments DARKNESS and MEC. MEC uses a stack-up of 20 fMESSI boards and it has been operating at the 8 meter Subaru telescope since 2018 [31]. 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), TESes (Transition Edge Sensors) connected to RF microchips for high-density frequency multiplexing. Since the analog inputs each have 2 GHz of bandwidth, the number of detectors per channel depends on the channel bandwidth and separation. A typical application [32] has 1000 MKID channels separated by 2 MHz. The QICK has several 1K multiplexed detectors/ADC and DAC, typically limited by the available FPGA logic. MKIDs and TESes coupled to microwave cavities require an excitation tone at the resonance frequency of the pixel, henceforth the DACs are used to provide the same number of tone excitations. Using the QICK as a MKID/TES controller will be described in a separate paper.

This system is simpler, more compact, and cheaper than the standard 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

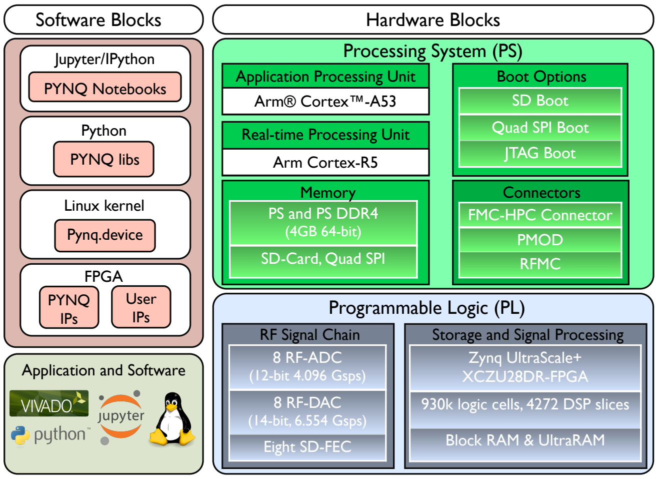


FIG. 2. The Xilinx XCZU28DR RFSoC chip block diagram. The chip consists of software blocks and hardware blocks. The software blocks can be subdivided into lower-level blocks such as the Linux kernel, and higher-level blocks such as the PYNQ software library whose functions are called from Jupyter notebooks.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

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 AXI interfaces and avoid the use of high-power drivers needed with external A/D and D/A devices that require LVDS or JDEM interfaces.The RFSoC also contains several different ARM processors, 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 SMA connectors. The QICK only requires a 50 watt, 12 volt 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 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. Fig. 3 shows the high-level block diagram of the controller which will be described in detail in Section III.

The RF board extends the eight RFSoC DAC outputs to either RF or DC coupled amplification and filtering (Fig. 5). 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 upconverts the RFSoC DAC output signal using the MM1-0212HSM double balanced mixer, which operates between 2-12 GHz. The RF board's amplifying chain has been optimized to have the highest performance between 3-8 GHz (Fig. 4). The output of the mixer (typically $\sim$ -23 dBm) is amplified by 40 dB and attenuated by two digital step attenuators. The step attenuators

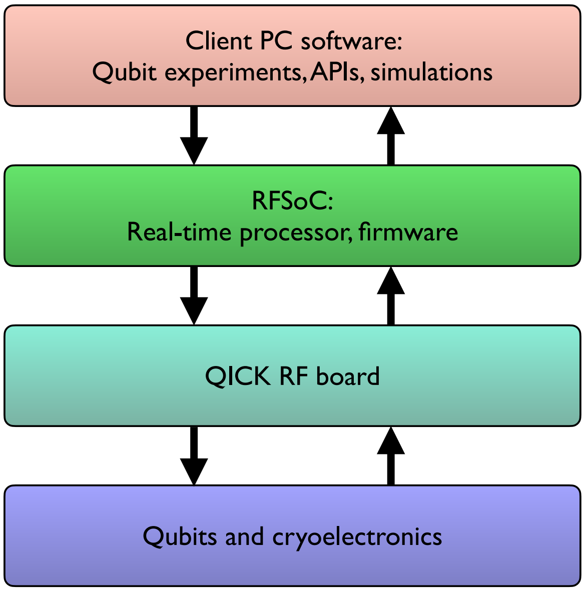

Client PC software:
Qubit experiments, APIs, simulations
RFSoC:
Real-time processor, firmware
QICK RF board
Qubits and cryoelectronics

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 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.

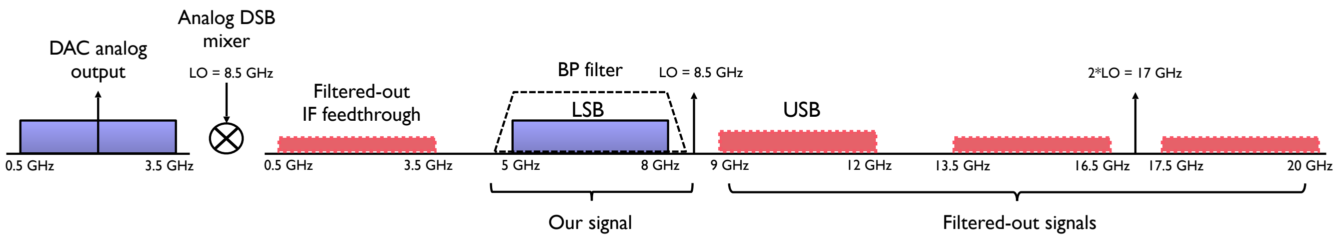


FIG. 4. The QICK RF board upconversion schematic. In this example, the QICK RF board LO is parked at 8.5 GHz. The user can place the IF signal anywhere in the 0.5-3.5 GHz band and then it is upconverted into the 5-8 GHz band. The QICK RF board filters the spurs in the unwanted sidebands.

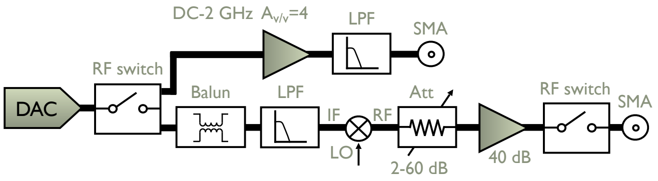


FIG. 5. The QICK RF board DAC output schematic. In software, the user can select the DAC output to be RF or DC coupled. The output signal is then processed by the associated DAC output chain.

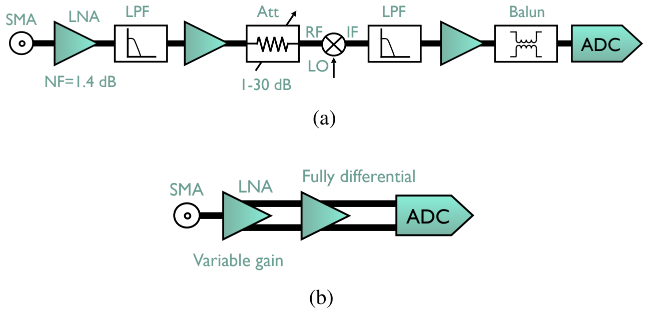


FIG. 6. The QICK RF board ADC input schematic. The QICK has four dedicated RF inputs (a), optimized for 4-8 GHz, and four dedicated DC coupled inputs (b) with a bandwidth of 1.5 GHz.

introduce a minimum insertion loss of 1 dB and total a maximum attenuation of up to 60 dB in 0.25 dB steps. So, 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 [33] or for the control of other quantum systems which integrate fast voltage or flux pulses, such as spin qubits in quantum dots [34].

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. So, 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, 10 qubits could be simultaneously read out by the same RF ADC with room to spare.

The RF bias board also provides 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 flux or voltage such as flux qubits or spin qubits in quantum dots. The bias maximum output voltage is ± 10 volts with 1 ppm resolution, 1 ppm INL, 7.8 nV $/\sqrt{\text{Hz}}$ 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 PLL and fractional frequency synthesizers. Each PLL has two outputs which are split in two by a Mini-Circuits EP2C+. 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 $\sim$ 1 GHz and up to 10 GHz or more. The high frequency (10 kHz-10 MHz) integrated jitter is 55 fs. The 10 Hz-10 MHz integrated jitter is $\sim$ 500 fs.

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. 7. The PS of this UltraScale+ device is a ZYNQ system with its own DDR4 memory, running 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

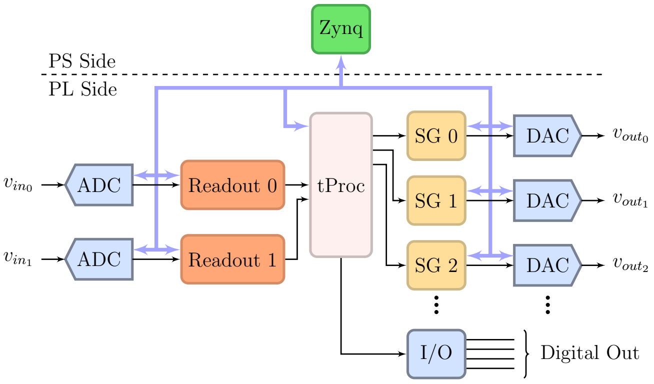


FIG. 7. The QICK firmware block diagram. The QICK firmware mainly consists of the timed-processor (tProcessor) block (Figure 8), the signal generator (SG) blocks (Figure 9) and the readout blocks (Figure 10). Instructions are passed between the RFSoC Zynq processor and the firmware blocks. These instructions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

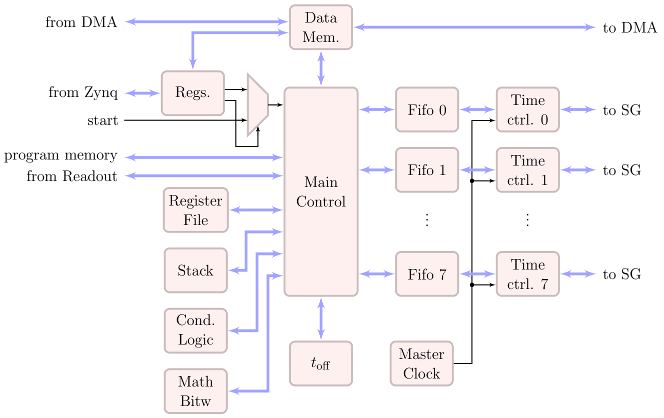


FIG. 8. The microsequencer (tProcessor) firmware block diagram. The tProcessor has a master clock and a time offset register $t_{\rm off}$ that is used to align user instructions in time. Timed instructions are dispatched to the signal generator (SG) (Figure 9) and readout blocks (Figure 10). Also, the user can execute common operations like register read/write, addition and substraction, 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.

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 in 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 [35]. 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 DDS blocks for digital upconversion. The output of each ADC is connected to a readout block, which implements fast digital downconversion, and 8 times decimation and filtering, followed by averaging and buffering for signal detection. An additional digital output I/O channel carries digital markers to control the signal generator and readout blocks, and is routed to physical connectors to allow 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 if the tProcessor is programmed with a low-level custom defined assembly language, high-level Python classes are provided to ease the description of the experiment. The QICK Github repository and QICK documentation website provide demonstrations and documentation of these classes [2, 35]. This allows users to directly access all of the capabilities of the system in a user-friendly environment.

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 branchings, register access, signal generation, and readback. The tProcessor is programmed and controlled with Python APIs.

To allow precise timing and control of qubit experiments, this block has the timed instruction infrastructure. When an instruction with a time tag is found in the program memory, this instruction is dispatched into the corresponding queue of the output channel. 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 channels gets full, the processor will wait until

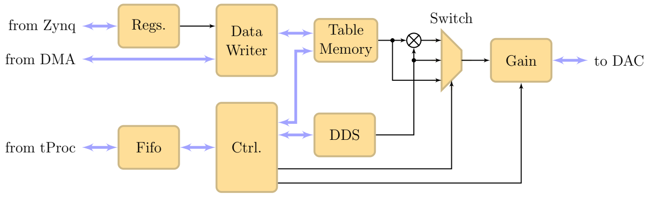


FIG. 9. The signal generator firmware block diagram. The user sends waveform parameters via Python (running on the Zynq PS) to the signal generator block. These waveforms are played at the time specified by the tProcessor block (Figure 8) to ensure phase coherence. The table memory block stores the I 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.

space becomes available before going further in the program.

The master clock is implemented using a 48-bit counter. This means that the user can run a single tProcessor program for up to eight days continuously. This number could be further increased to allow for longer experiments. 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 describing 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 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 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 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. The waveform parameters allow selection of the start address of the envelope, DDS frequency and phase, pulse duration, output selection and gain. This allows, for example, the output to switch efficiently from a Gaussian pulse of 20 ns duration and 2 GHz of frequency, to a pulse of 1 μs and no high-frequency modulation, and then back to a square envelope pulse at 2.8 GHz. The resolution for the fast DDS is 32-bit, which gives $\sim$ 1.5 Hz of resolution with a sampling frequency of 6 GHz on the output DAC.

An important feature of the signal generator block is phase coherence. For proper control of qubits, the signal generator 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 set of sine wave generators that never stop, and are switched to generate the output pulse.) To ensure phase coherence, the signal generator block computes the required phase when a frequency is set on the pulse.

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. This block provides 8 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 8 individual DDS blocks to allow covering the entire ADC bandwidth in the downconversion process. The switch determines which values are passed on: "product" for the downconverted signal; "input" for the input signal, and "DDS" for the values from the DDS blocks.

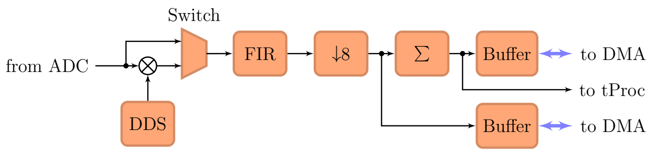


FIG. 10. The readout firmware block diagram. Incoming samples come at high-speed from the ADC block to the readout block. Digital downconversion is performed by multiplying the high-speed incoming samples with a fast DDS. The switch determines which mode of downconversion to do. The resulting signal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

After frequency-shifting the signal, the signal is low-pass filtered and decimated by 8. The decimation factor and filtering were selected for particular experiments but could be modified in future versions of the firmware to tailor specific needs. The decimated signal is fed into the average and buffer blocks. The buffer is mostly used for debugging as it captures the raw, decimated samples. The average block is configured to accumulate the sum of the decimated samples. 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, with the block automatically storing the results in the next memory address like in a circular buffer. Data from this buffer can be retrieved by using fast DMA transfers.

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. Special instruction on the tProcessor allow reads from this external port (real/imaginary), which is then used in a conditional branch statement. When designing experiments that use this feature, care must be taken so that the readout has completed before accessing the value.

IV. PERFORMANCE CHARACTERIZATION

Here we discuss the RF performance of the signal generator output. Fig. 4 shows the schematic of mixer upconversion from IF to RF using the RF bias board. Since the digitally generated IF can cover 3 GHz of bandwidth in each 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 LPF 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 to 8.5 GHz, the user can place pulses in a 3 GHz band without generating harmonics, and having enough LO suppression. Fig. 11 shows a demonstration of the clean mixer upconversion spectrum (with the RF board's additional >8 GHz LPF 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 LPF or BPF 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 does not use analog external IQ mixers; instead, it uses high frequency DDS and digital IQ mixers in the FPGA. There is no gain error other than a round off bit that introduces less than -100 dBm of error. So, unlike most qubit controllers, when the QICK is used with its RF board it 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 ± mIF for integers n and m. Fig. 4 shows how the QICK RF board filters all undesired outputs. As mentioned in Section 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. 4 before the mixer symbol). For instance, as in Fig. 4 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 GHz and 9-12 GHz. The IF and LO feedthrough, and nLO ± mIF 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 to 8.5 GHz band enables us to place our upconverted signal in the 4 to 8 GHz band. Fig. 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-10 MHz.

We also measured gain stability of the QICK versus time for a total of 16 hours. 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 and the board was sitting at room temperature with no additional temperature control. The gain was found to be stable with an RMS value of 0.07%.

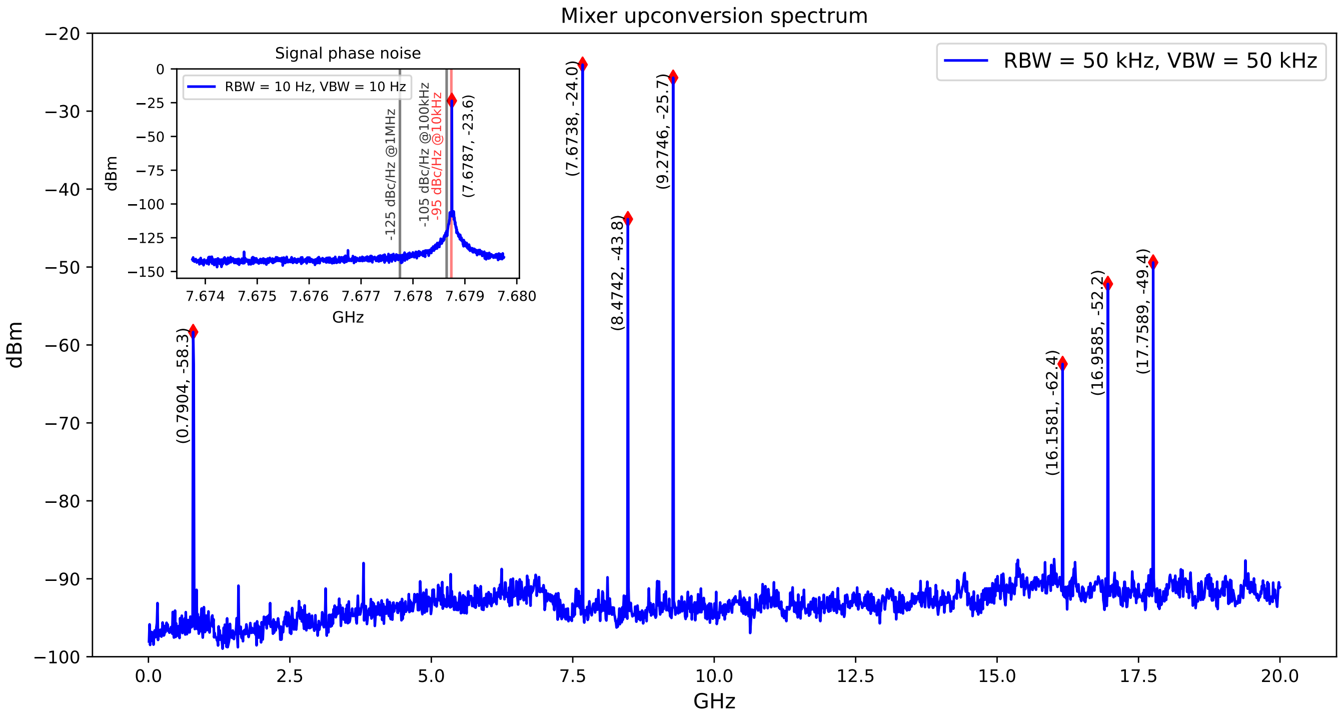


FIG. 11. QICK RF board performance: spectrum of mixer upconversion from IF to RF; inset is signal phase noise. In this measurement the RF board's additional >8 GHz LPF was disabled so as to display the full upconversion spectrum. Here an 0.8 GHz IF has been upconverted with an on-board LO set to 8.478 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.

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

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

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 Section 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. Fig. 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 to 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 to the ADC. When the upconversion and downconversion are enabled, the round trip latency increases to 113 ns (58 ILA clocks). When the x8 interpolation/decimation and filters 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 fed-back 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 ns to 211 ns depending on the DAC and ADC configuration.

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 [36–45]. 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

Sampling freq. (MHz)	Functions	Clocks	Latency
ADC:4096 DAC:6144	ADC and DAC with all digital features bypassed	46	90 ns
ADC:4096 DAC:6144	ADC and DAC with NCO enabled	58	113 ns
ADC:4096 DAC:6144	ADC and DAC, NCO enabled, and interpolation/decimation enabled	60	117 ns
ADC:4096 DAC:6144	Conditional evaluation and address jump	16	42 ns
ADC:4096 DAC:6144	Next pulse latency if address jump=TRUE	20	52 ns

TABLE II. The QICK firmware latency measurements.

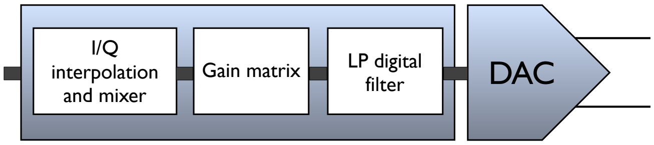
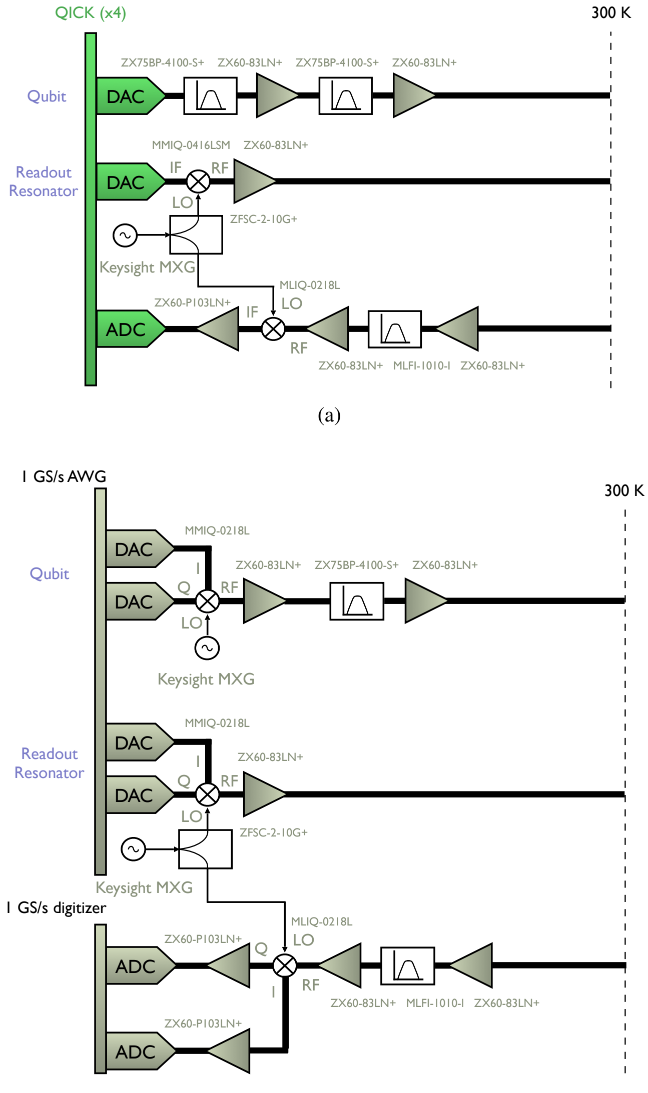


FIG. 12. Simplified RFSoC DAC module block diagram. The ADC block diagram is similar but the interpolator block is replaced by a decimator block.

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 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. Also, the QICK RF board's efficient filtering of the warm readout electronics was designed to enhance the readout signal's SNR.

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 [46]. Repeated QND measurements of the qubit yield the same result. If the readout electronics and the readout cavity are broadband (i.e. several GHz), several qubits 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.



(a)

(b)

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. Also, 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 Figure 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.

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 but has not yet been implemented in an experiment. Our eventual

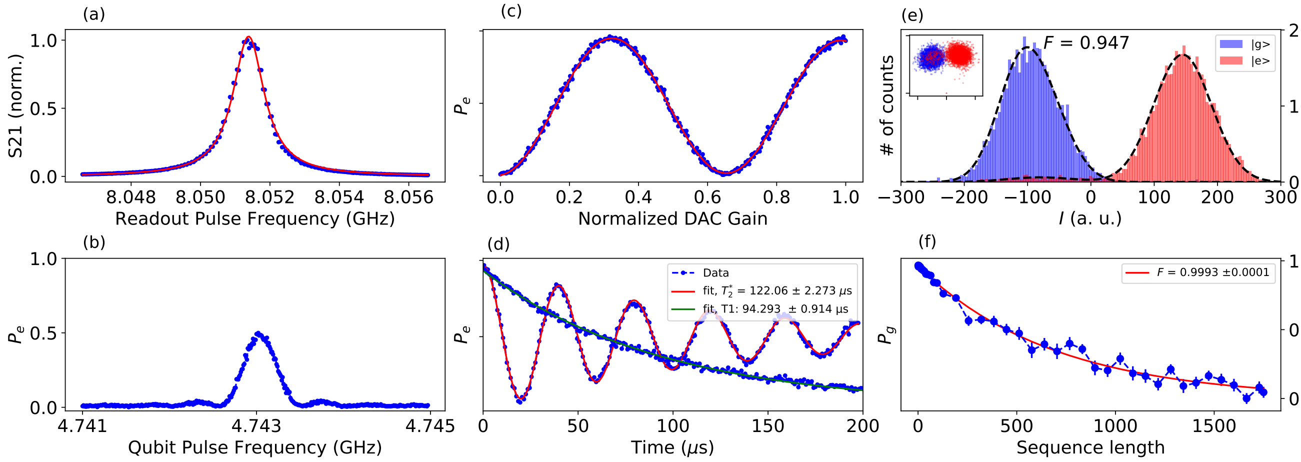


FIG. 14. The QICK was used to characterize a transmon qubit dispersively coupled to a readout cavity. Figures (a) and (b) show readout cavity and qubit spectroscopy measurements, and figure (c) shows qubit Rabi oscillations. Figure (d) show qubit T_1 and T_2 measurements (of 94 μs and 122 μs, respectively). 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 the standard RB averaged over 30 unique random sequences with 3000 shots per data point. The average gate fidelity is $\mathscr{F}_{avg} = 99.93\% \pm 0.01\%$ which approaches the estimated coherence-limited gate fidelity of $\mathscr{F}_{lim} = 99.96\%$. All measurements were performed entirely on the QICK, by running a Jupyter notebook on-board which compiled experiments and acquired data.

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 standard qubit controller used in the Schuster lab, the QICK required half the DAC channels and half the amount of analog upconversion (Figure 13).

The QICK was used to control and read out a 3D transmon qubit that is being used as a dark matter detector [30]. 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 $\Delta\phi = \omega * \tau$, 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 (I, Q) readout values were acquired for 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). Lastly, we characterize the fidelity of our single-qubit gates through standard randomized benchmarking (RB). 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 $\pm Z/2$ 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 $\mathscr{F}_{avg} = 99.93\% \pm 0.01\%$ as shown in Fig. 14 (f). The coherence-limited gate fidelity is estimated to be $\mathscr{F}_{lim} = 99.96\%$. Interested readers may follow excellent review articles describing standard qubit characterization experiments [47, 48].

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, a technique that the lab recently demonstrated in [33]. 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 [2].

Our future work focuses on hardware, firmware and software improvement and improved functionality. On the hardware side, we will explore the use of the Xilinx ZCU216 board hosting a ZU49DR RFSoC Gen3 FPGA [49]. The ZU49DR DACs run at 10 GS/s, with increased analog bandwidth, > 6 GHz. Some of the qubit experiments could be run without the need of an external analog mixer, 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 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 [50, 51], which will help developers.

VI. ACKNOWLEDGEMENTS

This manuscript 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 National Quantum Information Science Research Centers, Quantum Science Center. This work was funded in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730449. This work was supported by the Army Research Office under Grant No. W911NF1910016. S Sussman is supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. A Agrawal is supported by the Heising-Simons Foundation. The Fermilab team thanks Gaston Gutierrez (Fermilab) for his help bridging the gap between engineering and quantum mechanics. The authors thank National Quantum Information Science Research Centers Q-NEXT, SQMS, and C2QA members who participated in discussions.

[1] "Xilinx and Avnet ZCU111 evaluation board user guide," `https://www.xilinx.com/support/documentation/boards_and_kits/zcu111/ug1271-zcu111-eval-bd.pdf` (2020).

[2] "QICK Github repository," `https://github.com/openquantumhardware/qick` (2021).

[3] P. W. Shor, SIAM Journal on Computing **26**, 1484–1509 (1997).

[4] C. Gidney and M. Ekerå, Quantum **5**, 1 (2021), arXiv:1905.09749.

[5] C. H. Bennett and G. Brassard, Theoretical Computer Science **560**, 7–11 (2014).

[6] 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** (2020), 10.1038/s41586-020-2401-y.

[7] 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).

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195–202 (2017).

[9] 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," (2021), arXiv:2009.11482 [quant-ph].

[10] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Annual Review of Condensed Matter Physics **11**, 369–395 (2020).

[11] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi, and et al., Nature Nanotechnology **13**, 102–106 (2017).

[12] R. Schirhagl, K. Chang, M. Loretz, and C. L. Degen, Annual Review of Physical Chemistry **65**, 83 (2014), pMID: 24274702, https://doi.org/10.1146/annurev-physchem-040513-103659.

[13] I. Bloch, Nat Phys **1**, 23 (2005).

[14] L. M. Sager, S. E. Smart, and D. A. Mazziotti, Physical Review Research **2** (2020), 10.1103/physrevresearch.2.043205.

[15] 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** (2019).

[16] S. Asaad, C. Dickel, N. K. Langford, S. Poletto, A. Bruno, M. A. Rol, D. Deurloo, and L. DiCarlo, npj Quantum Information **2** (2016), 10.1038/npjqi.2016.29.

[17] T. Walter, P. Kurpiers, S. Gasparinetti, P. Magnard, A. Potočnik, Y. Salathé, M. Pechal, M. Mondal, M. Oppliger, C. Eichler, and et al., Physical Review Applied **7** (2017), 10.1103/physrevapplied.7.054020.

[18] 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).

[19] "ARTIQ Website," `https://m-labs.hk/experiment-control/artiq/` (2021).

[20] C. A. Ryan, B. R. Johnson, D. Ristè, B. Donovan, and T. A. Ohki, Review of Scientific Instruments **88** (2017), 10.1063/1.5006525.

[21] "Keysight Quantum Solutions," `https://www.keysight.com/us/en/solutions/emerging-technologies/quantum-solutions.html` (2021).

[22] "Zurich Instruments Website," `https://www.zhinst.com/americas/en` (2021).

[23] "Quantum Machines Website," `https://www.quantum-machines.co/` (2021).

[24] Y. Xu, G. Huang, J. Balewski, R. Naik, A. Morvan, B. Mitchell, K. Nowrouzi, D. I. Santiago, and I. Siddiqi, "Qubic: An open source fpga-based control and measurement system for superconducting quantum information processors," (2021), arXiv:2101.00071 [quant-ph].

[25] W. D. Kalfus, D. F. Lee, G. J. Ribeill, S. D. Fallek, A. Wagner, B. Donovan, D. Riste, and T. A. Ohki, IEEE Transactions on Quantum Engineering **1** (2021), 10.1109/tqe.2020.3042895.

[26] S. W. Jolin, R. Borgani, M. O. Tholén, D. Forchheimer, and D. B. Haviland, Review of Scientific Instruments **91**, 124707 (2020).

[27] "Xilinx RFSOC Website," `https://www.xilinx.com/products/silicon-devices/soc/rfsoc.html` (2021).

[28] "Python productivitye for zynq," `https://pynq.io` (2021).

[29] "Zynq Ultrascale+ RFSOC Product Data Sheet: Overview (DS889)," `https://www.xilinx.com/support/documentation/data_sheets/ds889-zynq-usp-rfsoc-overview.pdf` (2020).

[30] A. V. Dixit, S. Chakram, K. He, A. Agrawal, R. K. Naik, D. I. Schuster, and A. Chou, Physical Review Letters **126** (2021), 10.1103/PhysRevLett.126.141302.

[31] 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, Publications of the Astronomical Society of the Pacific **132** (2020), 10.1088/1538-3873/abc60f.

[32] 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, Review of Scientific Instruments **91** (2020), 10.1063/5.0029457.

[33] H. Zhang, S. Chakram, T. Roy, N. Earnest, Y. Lu, Z. Huang, D. K. Weiss, J. Koch, and D. I. Schuster, Physical Review X **11** (2021), 10.1103/PhysRevX.11.011010.

[34] R. Hanson, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen, Reviews of Modern Physics **79**, 1217 (2007).

[35] "QICK documentation website," `https://qick-docs.readthedocs.io/en/latest/` (2021).

[36] 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, Physical Review A - Atomic, Molecular, and Optical Physics **76** (2007), 10.1103/PhysRevA.76.042319.

[37] J. Gambetta, A. Blais, D. I. Schuster, A. Wallraff, L. Frunzio, J. Majer, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Physical Review A - Atomic, Molecular, and Optical Physics **74** (2006), 10.1103/PhysRevA.74.042318.

[38] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai, Nature **398** (1999), 10.1038/19718.

[39] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Physical Review A - Atomic, Molecular, and Optical Physics **86** (2012), 10.1103/PhysRevA.86.032324.

[40] A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, J. Majer, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Physical Review Letters **95** (2005), 10.1103/PhysRevLett.95.060501.

[41] J. M. Martinis, Quantum Information Processing **8** (2009), 10.1007/s11128-009-0105-1.

[42] J. Preskill, Annual Symposium on Foundations of Computer Science - Proceedings (1997).

[43] D. P. DiVincenzo, Fortschritte der Physik **48** (2000), 10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E.

[44] R. J. Schoelkopf and S. M. Girvin, Nature **451** (2008), 10.1038/451664a.

[45] M. H. Devoret and R. J. Schoelkopf, Science **339** (2013), 10.1126/science.1231930.

[46] D. I. Schuster, *Circuit Quantum Electrodynamics*, Ph.D. thesis, Yale University (2007).

[47] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, Applied Physics Reviews **6**, 021318 (2019), https://doi.org/10.1063/1.5089550.

[48] M. Naghiloo, "Introduction to experimental quantum measurement with superconducting qubits," (2019), arXiv:1904.09291 [quant-ph].

[49] "Xilinx and Avnet ZCU216 evaluation board user guide," `https://www.xilinx.com/support/documentation/boards_and_kits/zcu216/ug1390-zcu216-eval-bd.pdf` (2020).

[50] "IBM Qiskit library," `https://qiskit.org/` (2021).

[51] "IBM Qiskit OpenQASM library," `https://github.com/Qiskit/openqasm` (2021).