A Modular RFSoC-based Approach to Interface Superconducting Quantum Bits

Richard Gebauer, Nick Karcher, and Oliver Sander
Institute for Data Processing and Electronics
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
richard.gebauer@kit.edu

***Abstract*—Quantum computers will be a revolutionary extension of the heterogeneous computing world. They consist of many quantum bits (qubits) and require a careful design of the interface between the classical computer architecture and the quantum processor. Even single nanosecond variations of the interaction may have an influence on the quantum state.**

In this paper, we present the modular design of the FPGA firmware which is part of our qubit control electronics. It features so-called digital unit cells where each cell contains all the logic necessary to interact with a single superconducting qubit. The cell includes a custom-built RISC-V-based sequencer, as well as two signal generators and a signal recorder. Internal communication within the cell is handled using a modified Wishbone bus with custom 2-to-N interconnect and deterministic broadcast functionality. We furthermore provide the resource utilization of our design and demonstrate its correct operation using an actual superconducting five qubit chip.

***Index Terms*—Data acquisition, FPGA, Pulse generation, Quantum bits, Quantum-classical interface, Quantum computing, RFSoC, RISC-V**

I. Introduction

Quantum computing promises many applications, ranging from quantum encryption [1] and simulation [2], over drug research [3] and material science [4], to quantum machine learning [5] and other optimization problems [6]. At the same time, quantum computers are no general purpose machines and are best used as accelerators in a heterogeneous computing cluster. To build a useful quantum computer, many quantum bits (qubits) need to be combined into a quantum processor [7]. Such quantum processors are not able to execute programs autonomously like regular processors. They require external control, implemented in classical architectures and connected to them via a so called quantum-classical interface. This interface also handles the data exchange with the quantum processor. It thus plays a central role in the architecture of a quantum computer.

The requirements for such an interface strongly depend on the specific qubit realization. For qubits made out of superconducting circuits, like Transmon qubits [8], microwave pulses with arbitrary shape and gigahertz frequencies have to be generated and acquired. Timing of these pulses is crucial with nanosecond accuracy necessary to obtain reproducible results. Qubit state measurements involve microwave signal recording and demodulation. Performing computational tasks requires executing well-defined sequences of control pulses and state measurements. These pulses and measurements act as single gate operations. Multiple such operations are concatenated to perform the quantum algorithm [9]. For the pulses, not only their duration and shape matter but also their exact frequency, amplitude and phase. The delays between multiple pulses can also have a significant impact on the results. For some operations, it is furthermore necessary to perform conditional pulses depending on the result of a previous state measurement [10], [11]. With single gate operations on the order of tens to hundreds of nanoseconds, response latencies should also be on the same order of magnitude.

Full-scale quantum processors do not yet exist and their design is still considered to be fundamental research in physics [12], [13]. While scaling up is one research direction, scientists also strive to improve the properties of single qubits, like minimizing the error rate [14], [15]. To perform experiments in both research contexts, we developed a versatile control electronics based on a heterogeneous radio-frequency system-on-chip (RFSoC) device that implements a quantum-classical interface tailored for superconducting qubits. Our platform fully satisfies the major requirements: control of pulse properties down to nanosecond precision, high-level programmability in quantum computing frameworks, modularity, scalability, and flexibility.

In this paper, we detail the design implemented within the programmable logic (PL) of this system. To provide scalable control and readout, the design is structured in a similar way as the quantum processor. A digital unit cell contains all necessary capabilities to interact with one qubit. This building block is then instantiated multiple times to provide individual control over up to 15 qubits with a single RFSoC.

II. Related Work

In the field of research of superconducting qubits, general-purpose laboratory equipment is widely used to generate and analyze microwave pulses. With increasing system complexity, utilizing these devices becomes unfeasible due to significant communication delays, bad scaling properties and high relative cost. Therefore, FPGA-based systems emerged being individually designed for certain experiments to meet the high data processing and latency demands, e.g. [10], [11], [16], [17].

Recently, commercial products have appeared on the market that specifically target superconducting qubits. Noteworthy products are the OPX of Quantum Machines [18], the Quantum Computing Control System (QCCS) of Zurich Instruments [19], and the Quantum Engineering Toolkit (QET) of Keysight [20]. All of these systems offer the sequencing, generation and detection of base-band microwave pulses on one or multiple FPGAs. Based on their data sheet [20], Keysight's QET uses different hardware modules for signal generation and digitization that are combined in a PXIe chassis. Zurich Instruments' QCCS also distributes the necessary capabilities over different devices suggesting higher latencies than if integrated on a single SoC. None of these two systems uses a comparable structure to the digital unit cell proposed in this paper. Signal generation and acquisition are distributed over physically separated devices. In contrast, Quantum Machines' OPX integrates them on a single device. They utilize "pulsers" that encapsulate the functionality for a single qubit, similar to the digital unit cell in this paper. A more detailed comparison is impossible as the internal technical details are not publicly available for these products.

III. Interfacing Superconducting Quantum Bits

Qubits are the building blocks of a quantum processor. Similar to a classical bit, they have two fundamental states, labeled $|0\rangle$ and $|1\rangle$. Yet, they can also stay in an arbitrary superposition $|q\rangle = \alpha |0\rangle + \beta |1\rangle$ with $\alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$. The state of the qubit can then be depicted as a point on the surface of a sphere, called Bloch sphere, with the states $|0\rangle$ and $|1\rangle$ being located at the north and south pole, respectively. In the typical frame, they are located on the Z axis while the X and Y axis span the plane of the equator. Accordingly, operations on the qubit are rotations around the surface of the sphere.

Superconducting qubits are microscopic, non-linear resonance circuits fabricated with similar methods as used in the semiconductor industry. They are made from superconducting material and exhibit quantum behavior when cooled below the transition temperature where the material completely drops its electrical resistance. This can be employed to engineer systems showing two fundamental quantum states used as computational subspace. Due to the low temperature requirement of typically tens of millikelvin, the chip with the quantum bits is located inside a cryostat.

To control a qubit's state, microwave pulses with gigahertz frequency and nanosecond time resolution are used. Each qubit has a dedicated transition frequency f_{01} corresponding to the energy difference of the two basis quantum states. When irradiated with a microwave pulse at this frequency, the state will oscillate around the Bloch sphere between $|0\rangle$ and $|1\rangle$, called Rabi oscillation. By adjusting either the duration or the amplitude of the pulse, the rotation angle around the Bloch sphere can be varied. Changing the phase of the microwave pulse will change the axis of rotation in the equatorial plane of the sphere. As the phase determines the frame of reference for the sphere, one can also adjust the global phase to perform a virtual Z rotation around the equator. As result, one can perform arbitrary rotations. Depending on the actual superconducting circuits, additional current pulses might be required to perform special single or two-qubit gates.

The readout of a qubit is commonly performed as dispersive readout where an additional microwave resonator is coupled to the qubit. Depending on the qubit state, the resonator will experience a slight shift in its resonance frequency f_r. By probing the resonator with a microwave pulse near this frequency, the qubit state will be encoded in the amplitude and phase response. Due to the low temperature requirement, the chip needs to be shielded as best as possible from the surrounding. Thus, the probe pulse will be dampened inside the cryostat to thermalize the noise present on the signal. After interacting with the resonator with a strength of only a few photons, it needs to be amplified again to be detectable by the room-temperature electronics. Special low-noise amplifiers are required to obtain a signal-to-noise ratio (SNR) good enough to extract the qubit state from a single measurement. If such an amplifier is not available, multiple repetitions can still be performed and the results averaged to increase the SNR. However, in this case, the qubit state from individual measurements cannot be determined. More detailed information on superconducting qubits can e.g. be found in [9].

IV. System Architecture

The system we developed aims to facilitate these control and readout mechanisms to provide a quantum-classical interface for superconducting qubits (see Fig. 1). It is based on the heterogeneous architecture of a Xilinx Zynq UltraScale+ RFSoC. The chip incorporates an FPGA, a quad-core ARM Cortex-A53 application processor (APU), a dual-core ARM Cortex-R5 real-time co-processor (RPU), as well as eight multi-gigasample AD and DA converters (ADCs/DACs). The converters operate at $4\,\mathrm{GHz}$ sampling frequency and handle the microwave generation and digitization in the complex base-band. We utilize decimation and interpolation filters to obtain a per-channel data rate of $1\,\mathrm{GSPS}$ within the programmable logic (PL). The signals are digitally represented as in-phase and quadrature (I/Q) components, each with $16\,\mathrm{bit}$ resolution. Hence, two converter channels per signal input and output are required to handle both quadratures representing the complex-valued base-band. A separate radio-frequency (RF) frontend electronics with I/Q mixers and microwave sources translates these base-band signals from and to the frequency range of superconducting circuits, typically in the order of 4 to $10\,\mathrm{GHz}$.

The APU hosts a Yocto-based Linux operating system which initializes the platform at boot time. Then, it starts the modular ServiceHub framework [21] to provide means for communication and external configuration. For each type of PL module, a ServiceHub plugin exists facilitating user access to these modules. The RF frontend electronics can also be controlled by a ServiceHub plugin via SPI and I^{2}C. The communication to the user is based on remote procedure calls (RPC) and utilizes the open-source framework gRPC [22]. Thus, the client can be written in any language that is supported by gRPC. As many physics laboratories use Python, we provide a Python client for our platform. It integrates with

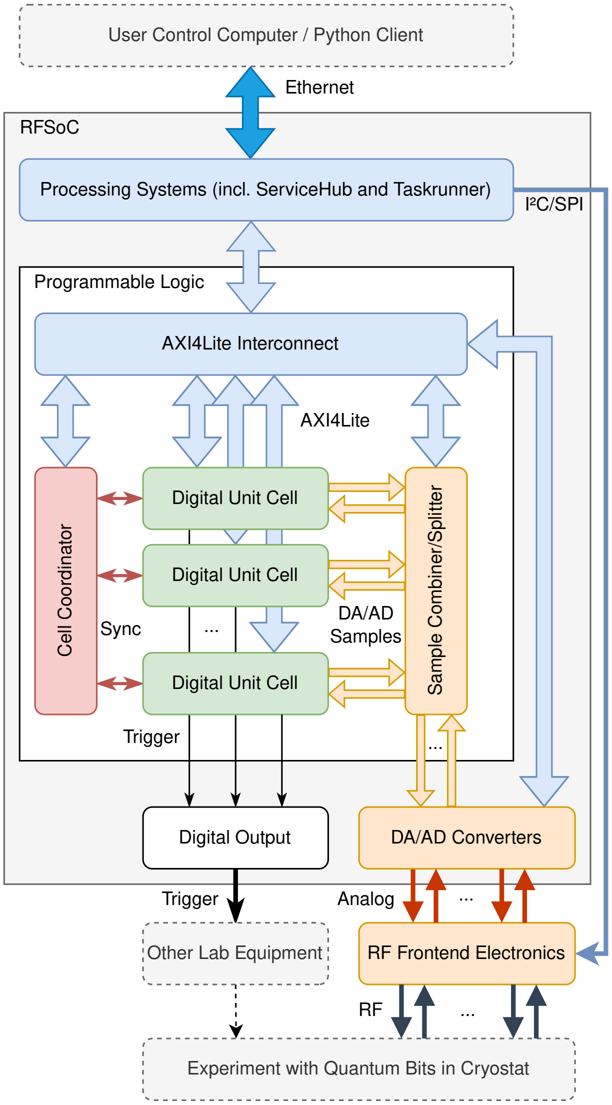


Fig. 1. Architecture of our heterogeneous electronics platform.

the open-source quantum measurement suite Qkit [23]. The RPU hosts the Taskrunner framework [24] which provides convenient access to the real-time processor. It complements the PL with versatile, low-latency real-time control, data aggregation and evaluation features. Both processors communicate with the PL using a register-based AXI4Lite bus where the modules are mapped into the physical memory address range of the PS. Access is performed by simple memory read and write operations using the AXI HPM FPD interface.

The user design in the PL operates on a single clock domain of $250\,\mathrm{MHz}$ directly derived from the converter clock. This avoids clock domain crossings and guarantees deterministic timing and nanosecond accuracy, which is crucial for the control of superconducting qubits. The digital unit cells provide the main functionality within the PL. Each one contains all the necessary modules to control and read out a single qubit, as presented in the following section. By implementing multiple such cells, multiple qubits can be individually controlled by a single system. To ensure synchronicity between the cells and facilitate inter-cell-communication, a special cell coordinator is connected to each of these cells which can start any subset of them simultaneously. The digital microwave signals generated in the unit cells are routed via AXI-streams to the DACs. In the simplest case, each digital unit cell is connected to separate converter channels for readout and control pulses. To reduce the channel count, multiple signals can also be frequency-division multiplexed and combined (added up) onto one output. Likewise, the returning digitized microwave signals from the ADCs can be distributed and split up onto the belonging cells. Besides the microwave pulses, also digital trigger signals can be generated. These can e.g. be used to trigger additional current sources which influence the experiment.

As this builds up to quite a complex system with a lot of dependencies, directly configuring all the modules is inconvenient and error prone. Instead, we provide a high-level experiment description language based on Python that can be used to functionally describe the cell's control flow and output. This will be automatically compiled into RISC-V instructions for the sequencers (see the following section) and the configuration parameters for the other modules within the different digital unit cells. Furthermore, an appropriate task to fetch the data from the PL and transfer it to the user is loaded onto the Taskrunner. The remaining configuration set is loaded into the PL modules. Then, the user starts the execution of the task in the Taskrunner which, in turn, will simultaneously start the sequencers in the relevant digital unit cells using the cell coordinator. After all sequencers have finished their execution, the Taskrunner fetches the data from the data storages. Depending on the configuration, it can also perform multiple repetitions and accumulate or average the resulting data. Finally, it transfers the data back to the user.

V. Digital Unit Cell

The core of our PL design is the digital unit cell which will be described in detail in this section. It contains all the relevant logic to generate and analyze pulses in the complex-valued base-band for a single qubit and to control peripheral devices for additional capabilities. We chose to implement the entire module on the PL because qubit control requires at least timing precision on the granularity of single clock cycles. Real-time capabilities of the RPU could be utilized but are impacted by latency and interferences on the AXI communication infrastructure of the PS.

A. Module Architecture

The architecture of the digital unit cell is depicted in Fig. 2. Most internal communication inside the cell is handled by a Wishbone (WB) bus [25]. WB was selected because of its simplicity and resource efficiency compared to a full featured AXI interface. We use a custom implementation to ensure deterministic timing, which is essential for our application. Because the same interface is used by the PS to configure the cell, we utilize an AXI4Lite to Wishbone bridge to translate

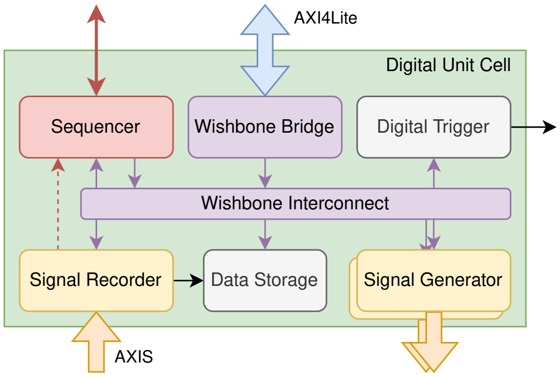


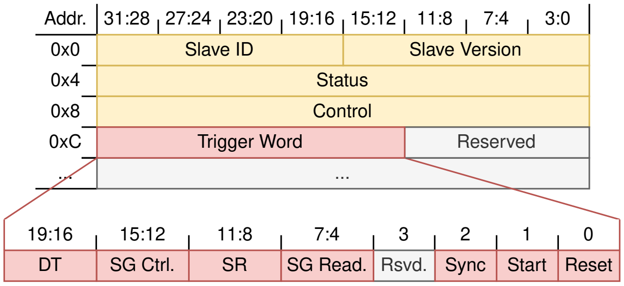
Fig. 2. Architecture of a single digital unit cell. Arrows of the WB bus indicate the direction from master to slave.

the register accesses for the internal bus. Two signal generators create the required pulses to control and read out the qubits utilizing an AXI-stream interface. A signal recorder takes the digitized signal from the ADCs and demodulates it to obtain the qubit state. A dedicated data storage can collect the resulting data from the signal recorder and the sequencer. A digital trigger block can generate digital signals to address and trigger external lab equipment. All modules are controlled and activated by the sequencer which orchestrates their execution in single-cycle steps (4 ns). The signal recorder directly reports all measured qubit states back to the sequencer which can then perform a fast conditional response. In all other cases, the modules communicate exclusively via the Wishbone bus.

B. Communication Infrastructure

The Wishbone bus inside the digital unit cell features a 16 bit address width and a 32 bit data width. A custom Wishbone interconnect allows for two masters and up to seven connected slaves. Both sequencer and WB bridge are connected as masters. In case of a conflicting access, the sequencer always takes priority in order to keep deterministic timing during executions. The WB bridge also performs an address translation from byte-based addressing as used by the AXI4Lite bus to register-based addressing used by WB. All slave modules have a special WB register interface implemented that guarantees a deterministic response time of 2 cycles without stalling. With the interconnect, a read operation takes exactly 4 cycles from the sequencer to return a result. Access from the WB bridge will take one extra cycle and might be stalled if the sequencer is currently accessing the bus. With the deterministic latency in mind, we modified the interconnect to allow a single pipelined register access each cycle on the bus even if a previous operation originating from one of the masters has not finished yet. This way, we can ensure that the sequencer can always issue trigger commands on the WB bus with deterministic access latency.

The multiplexing between the modules happens according to the highest three address bits. While `000` up to `110` represent the according connected module, `111` acts as broadcast modifier. In this case, the bus operation will be forwarded to all connected slave modules at the same time. We utilize this feature to provide means to trigger all the connected modules with a common trigger word at the same time without utilizing a separate trigger infrastructure. The register interfaces of all slave modules are therefore starting in a similar way with an info, a status and a control register (see Fig. 3). Afterwards, a broadcast register follows with a 20 bit trigger word field that can be strobed by a write access. The remaining registers can be freely and independently used depending on the demand of the modules. Special trigger commands are shared between all modules to reset them, mark the start of an execution, and to synchronize the NCOs inside the two signal generators and the signal recorder.



Addr.	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
0x0	Slave ID				Slave Version			
0x4	Status							
0x8	Control							
0xC	Trigger Word					Reserved		
...	...							

19:16	15:12	11:8	7:4	3	2	1	0
DT	SG Ctrl.	SR	SG Read.	Rsvd.	Sync	Start	Reset

Fig. 3. Common start of the register interfaces of all modules inside the digital unit cell. The address offset is given in bytes. Signal generators (SG), signal recorder (SR), and digital trigger (DT) are abbreviated.

C. Sequencer

The sequencer is the core of the digital unit cell. It controls all connected slave modules and can e.g. schedule pulses or start a recording. The user can define a sequence of operations in 4 ns steps using the RISC-V instruction set architecture (ISA). We chose the RISC-V ISA as it is state of the art, easily extensible, very flexible, hardware efficient, provides a rich ecosystem, and is well established in the scientific community. From the modular instruction sets of the RISC-V ISA, we implemented most of the base integer and multiplication set, as well as a custom special-purpose set for the sequencing. In total, 33 instructions are available for the sequencer as well as 32 registers. The following operations are part of the special-purpose set:

`TRIG`: Writes the given trigger word to the broadcast register of all connected WB slave modules.

`WAIT-IMM`: Delays the execution by the given number of clock cycles.

`WAIT-REG`: Delays the execution by the number of clock cycles given in the defined register.

`WAIT-REG-TRIG`: Same as `WAIT-REG` but reduces the wait time given in the register by one cycle. This can e.g. be used after a `TRIG` command to wait a register-defined time but include in it the duration of the previous command.

`SYNC-EXT`: Waits for external input before continuing with the program execution. This can e.g. be the resulting qubit state returned by the signal recorder.

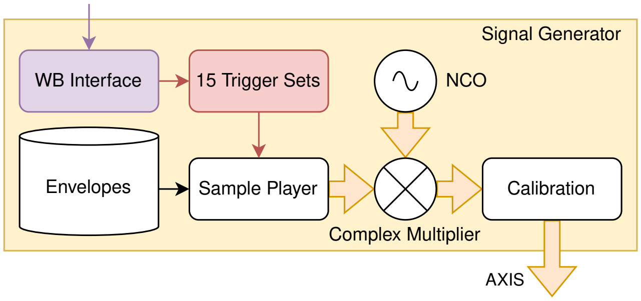


Fig. 4. Structure of the signal generator inside the digital unit cell.

SYNC-START: Ends the execution of the sequencer and returns to an idle state waiting for a new start command.

Most instructions are optimized to execute in only one cycle for highest performance. Only multiplication takes 6 cycles in order to relax the timing requirements for the 32 bit times 32 bit multiplication. Instructions entailing a jump in the program counter (branch instructions if the comparison yields true, as well as the unconditional JAL jump operation) take 3 cycles. The different wait operations take as long as specified in the command and the sync commands might wait an undetermined time on external input. Currently, up to 1024 instructions can be stored inside a BRAM. Typical experiments require tens of instructions to be executed. It is therefore enough for nearly all imaginable experiments but can also be easily extended by enlarging the BRAM, if necessary.

The sequencer has both a WB master and slave interface. As for every other module, the slave interface is used to configure and control the sequencer. The master interface can reconfigure the connected slave modules or fetch data from them. The according load and store operations will take 8 cycles as they wait for the response of the WB bus. 4 cycles account for the deterministic latency of the bus and 4 for processing the operation in the sequencer, applying the output to the bus and processing the return signals. Trigger commands are applied as pipelined block write operations as it is essential that they are issued each cycle without stalling. The sequencer does not wait for the bus return when applying trigger commands. Therefore, after finishing the block write at the sequencer, the bus might still be processing while the sequencer already executes the next operation. If this is a normal load or store operation, it will be delayed as the sequencer first has to wait until the bus is not busy anymore. This could be further optimized by always applying pipelined block write operations and exploiting the deterministic latency of our modified WB bus.

D. Signal Generator

Each digital unit cell has two signal generators, one for readout pulses and one for control pulses. It contains 15 trigger sets that can be selected by a 4 bit trigger command within the trigger word. Each trigger set represents a certain pulse that can be played by the signal generator. The following properties can be individually set for each trigger set:

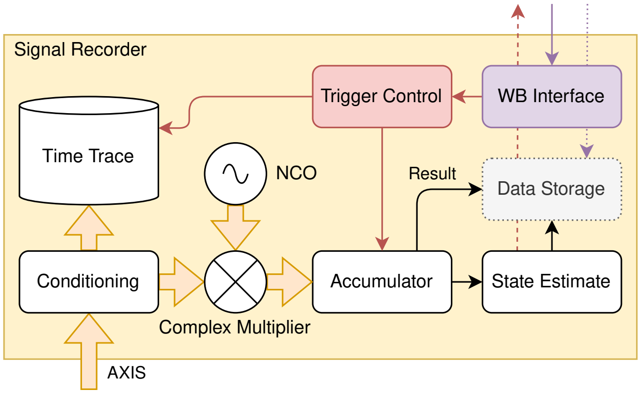


Fig. 5. Structure of the signal recorder inside the digital unit cell.

- The duration of the pulse in cycles.
- The phase offset of the pulse relative to a global phase reference inside the module.
- A scaling factor to change the amplitude of the pulse.
- Address offsets for I and Q envelopes inside the envelope memory. These can also point to the same address.
- Option to hold the last envelope value until another trigger is received. This is used for continuous wave operation and variable length pulse shapes like a trapezoid.
- Option to persist the phase offset in the global phase reference. This enables to perform a virtual Z rotation.

Additionally, the module has a common frequency reference which can be configured. It is implemented as a numerically controlled oscillator (NCO). Furthermore, an output calibration is possible to adjust the I and Q output amplitude.

The structure of the signal generator is presented in Fig. 4. All configuration as well as the trigger is fed to the module via its WB interface. When a trigger arrives and selects one of the 15 trigger sets (trigger value 0 is reserved for no operation), the configuration of this trigger set is loaded into the module and the execution is started. The sample player fetches the according I and Q envelope sample values from the envelope memory. It is 8 kB large and can therefore store 4096 real-valued samples (16 bit each) corresponding to up to about 4 µs of pulse data. As most pulses are in the order of tens to hundreds of nanoseconds, this is enough for most applications. The pulse will then be output by the sample player and fed into a complex multiplier. There, the envelope will be multiplied by the oscillating complex quadrature signal of the NCO to obtain the digital pulse in the base band. Afterwards, the I and Q quadratures can be independently calibrated and the signal leaves the module as AXI-stream.

E. Signal Recorder

The signal recorder obtains the digitized microwave signals from the converters and performs a digital down-conversion (DDC). Its structure is depicted in Fig. 5. Due to mixer and other imperfections in the analog setup, the raw I and Q data from the converters might not be balanced or have a 90° phase relation. To correct for this, a matrix multiplication will be performed on the raw input data:

$$\begin{pmatrix} I_{\text{out}} \\ Q_{\text{out}} \end{pmatrix} = M_{\text{cond}} \left[\begin{pmatrix} I_{\text{in}} \\ Q_{\text{in}} \end{pmatrix} - \begin{pmatrix} I_{\text{offset}} \\ Q_{\text{offset}} \end{pmatrix} \right] \tag{1}$$

Besides the 2×2 matrix M_{cond} to correct for amplitude and phase distortions, a DC offset can also be subtracted. The corrected raw time trace is stored inside a BRAM. It can later be used for debugging purposes or to visualize the raw input that was demodulated in the following. The signal will then be down-converted by a complex multiplier where it is multiplied with a reference oscillation having the negative frequency of the base band carrier. It thereby shifts the frequency of the signal carrier to DC. Afterwards, a low-pass filter and decimation are necessary to average the resulting I and Q component which are later used to determine the amplitude and phase response. In our case, we implement a boxcar integrator by using a simple accumulator to add up the samples over an adjustable time window. Alternatively, e.g. an FIR filter and decimation could be used for a better low-pass characteristic. At the same time, the accumulation yields a smaller latency to obtain a result.

While conditioning and complex multiplication are performed continuously, the boxcar integration as well as the storage of the raw time trace are only activated when the signal recorder receives a trigger signal. As the readout pulse experiences an electrical delay to the quantum chip and back, a trigger offset can be defined. Only after this offset time has passed, the trigger will be executed by the module. This way, the sequencer can trigger the readout signal generator and recorder at the same time and does not have to account for the electrical delay itself. Once the recording duration has passed, the accumulated result value is passed to the data storage, and used to estimate the qubit state.

For this estimation, the result is transformed into a binary information of `0` or `1` corresponding to the two possible qubit states that can be measured. This state result will be directly returned to the sequencer which can be programmed to wait for this value and store it in a register using the `SYNC-EXT` operation. The state will also be passed to the data storage where it can be aggregated and saved for later retrieval. As the data storage is tightly linked to the signal recorder, it is also shown in Fig. 5. For simple experiments, the signal recorder also provides an averaging functionality where obtained results will be summed up until the module is reset externally. This is especially helpful if a single measurement should be performed and repeated many times to obtain an averaged I and Q result value.

Different operation modes of the signal recorder can be distinguished, based on the received trigger value:

`RESET`: Resets all internal result data of the module.

`SINGLE`: Performs a single measurement.

`ONESHOT`: Performs a single measurement but does not forward it to the data storage. A typical use-case are two consecutive measurements where the first one is only used internally and will result in a state estimation on which the sequencer will react. The second one is then to obtain a measurement result of the experiment.

`CONTINUOUS`: Continuously performs consecutive measurements and returns the values to the storage module. This mode can be used to obtain a seamless stream of demodulated results without the need of the sequencer to trigger each single measurement. The continuous mode will continue until a `STOP` trigger is received. Together with the data storage and the Taskrunner, continuous operation over long periods is possible, e.g. to observe state changes of the qubit over time (so-called quantum jumps).

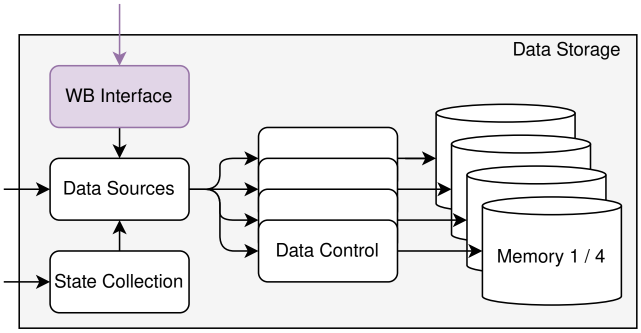


Fig. 6. Structure of the data storage inside the digital unit cell.

F. Data Storage

After demodulating the measurement results, the data needs to be persisted for later retrieval by the user. The data storage handles this in a configurable and flexible way. Its structure is shown in Fig. 6. The module contains four separated dual-port BRAMs to store values. These can be filled individually and in parallel. Thereby, result values can be partitioned in a user-defined way, e.g. to store both qubit states and I/Q results, or to store additional information from the sequencer in a separate BRAM. These memories provide an interface to consecutively append $32\,\text{bit}$ values to the memory until it is full. It furthermore contains an option to use the memory as circular buffer and wrap the address instead of rising an overflow flag if the memory is full. The second port of the BRAM is mapped into the WB interface for direct read and write access from sequencer and PS.

The signal recorder passes the single results and the estimated qubit states to the data storage. There, the qubit states will be concatenated to obtain $32\,\text{bit}$ words containing multiple of them. Depending on the estimation routine, one can either store 32 states if only one bit is used, or 10 states if also higher states are accounted for ($3\,\text{bit}$ information per state). Summarizing, the following data can be selected and stored inside the individual memories:

- Single I and Q result values
- Single estimated qubit states
- Concatenated qubit states (either 10 or 32 per register)
- Input data from the WB interface

TABLE I

Resource utilization on a Xilinx XCZU28DR RFSoC. Categories are configurable logic blocks (CLB), block RAMs (BRAM), and digital signal processing slices (DSP).

Entity	CLB	BRAM	DSP
Available resources	53 160	1080	4272
Full design	74.69 %	97.2 %	32.3 %
Single cell	(5.01 ± 0.17) %	6.48 %	2.15 %
Sequencer	(1.57 ± 0.03) %	0.09 %	0.09 %
Signal generator	(0.84 ± 0.05) %	2.04 %	0.47 %
Signal recorder	(1.45 ± 0.08) %	1.94 %	1.12 %
Data storage	(0.30 ± 0.02) %	0.37 %	0 %
Digital trigger	(0.43 ± 0.02) %	0 %	0 %
WB infrastructure	(0.63 ± 0.04) %	0 %	0 %

The last one is a special register in the WB interface to which the sequencer can write to append values to the memory blocks. This way, the sequencer can also perform calculations and store them or persist some additional values. The sequencer can also use the second port of the BRAMs mapped into the WB interface to have a memory extension, e.g. for arrays. Each memory block has a dedicated data control that decides which data source will be assigned to it. It also realizes the append and circular buffer logic as described above and provides status signals for the user, like empty, full, and overflow flags, as well as the current data size.

G. Digital Trigger

While the system covers most aspects to control and readout superconducting qubits, for some experiments it might be necessary to digitally trigger external measurement equipment for additional functionality. A common use case would be current pulses for special qubit gates. The digital trigger provides 15 trigger sets which can individually define which of the 8 available digital outputs should be activated and how many cycles it should stay asserted. A special option can be used for continuous activation. Each output can be individually inverted and a trigger offset specified. This is especially important to synchronize the action of external devices with the operation of the system.

VI. Results & Performance

We benchmarked our design using a Xilinx ZCU111 evaluation board with a custom-built analog frontend. The resource utilization of the complete design is provided in the following section. Each PL module is thoroughly unit tested during development and within a continuous integration workflow. We verified the correct operation of the complete design using the platform in a loop-back configuration and with an oscilloscope. Afterwards, we also performed experiments with actual superconducting qubits to show that operation in the field is working as expected. One exemplary experiment is presented below.

A. Resource Utilization

The resource utilization of the design is given in Table I for 15 digital unit cells. It is currently limited by the amount of available BRAMs, mainly due to the required resources for the NCOs (1.48 % per NCO) inside the signal recorders and generators. When more digital unit cells would be required, special Ultra RAM blocks available inside the RFSoC could be used in addition to normal BRAM blocks. For most experiments where 8 DAC output channels are sufficient, 15 unit cells are more than enough though. Yet, when utilizing another RFSoC with more channels, this optimization should be considered to allow for more unit cells on the system.

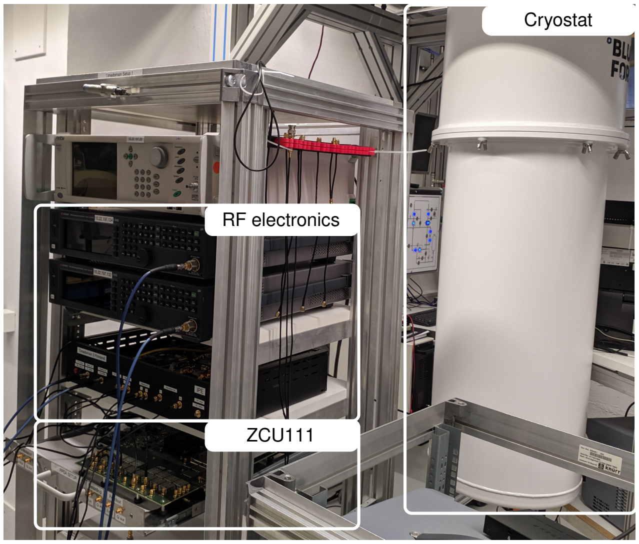


Fig. 7. Photograph of the experiment setup.

B. Functional Verification Utilizing a Five Qubit Chip

To test the system in the field, we used a well-characterized qubit chip with five superconducting Transmon qubits [26]. The qubits are not coupled to each other but via separate readout resonators to a common microwave line. Both control and readout pulses are fed into the chip on a single input line using frequency-division multiplexing. The complex-valued base-band frequencies for these signals range from $-260\,\mathrm{MHz}$ to $230\,\mathrm{MHz}$ by using in-phase and quadrature components. Typical control pulses are around $50\,\mathrm{ns}$ long. From the 15 available unit cells of our design, we utilize five, one for each qubit. In the RF electronics, we use two I/Q mixers with separate local oscillator microwave sources, one to up-convert all control pulses, and one for all readout pulses. For this, the readout signals of the five digital unit cells are digitally combined. The same applies to the control signals. After up-conversion, both RF signals are combined onto a single microwave line. By using a complex-valued base-band, the RF signals can be located flexibly on both sides of the local oscillator. A photograph to illustrate the experiment setup is given in Fig. 7.

As a first step, all five qubits need to be characterized and the necessary parameters to perform experiments determined. Our system can be used like a vector network analyzer to output a continuous tone with the readout signal generator and demodulate the response in the signal recorder. By changing the frequency of the internal NCO in both signal generator and recorder, a frequency sweep can be performed. As each qubit has its own unit cell, this can even be done simultaneously for all five qubits.

After the characterization of frequencies, signal amplitudes and pulse lengths, different experiments can be performed. One type of experiment is to determine the coherence time T_2 of the qubits. This is the characteristic timescale after which quantum information inside the qubit is lost due to external influences that disturb the quantum state. It can be measured using a Ramsey pulse sequence [9]. With the qubit starting in state $|0\rangle$, one performs a $\pi/2$ rotation around the X axis to end up on the equator of the Bloch sphere. There, one waits a variable delay before performing another $\pi/2$ rotation. When the control pulses are slightly detuned from the actual qubit frequency, by $5\,\mathrm{MHz}$ in our case, the state will oscillate around the bloch sphere during the waiting time. Then, the second $\pi/2$ rotation will not bring the qubit into the $|1\rangle$ state, but depending on the duration of the rotation to another state. This results in an oscillation between the $|0\rangle$ and $|1\rangle$ state with respect to the delay. In Fig. 8, the measurement result for the simultaneous measurement on all five qubits is shown. The state of the qubit is encoded in the phase response of the corresponding readout resonator due to the dispersive readout. One can see an exponential decay of the envelopes which is due to the finite coherence time as the information of the state is lost when waiting too long between the pulses. The decay constant is equivalent to the coherence time T_2. The values are extracted from damped sine fits and also given in the figure. They are within the expected range obtained from previous separate measurements with the same chip.

We also performed a set of other standard experiments that all yield results comparable to previous, consecutive measurements. However, the unit cell approach features intuitive parallel control and readout of all five qubits, thereby significantly reducing execution time by more than $80\,\%$.

VII. Conclusion

We presented the modular FPGA firmware of our RFSoC-based qubit control electronics. Our design is based on a digital unit cell that contains all necessary logic and capabilities to interact with a single qubit. All components of the cell, including the RISC-V-based sequencer, implement the required deterministic latency and support high-level programmability. Our custom Wishbone bus implementation contains additional features such as broadcasts for versatile synchronous triggering. Using a Xilinx XCZU28DR RFSoC, we can implement 15 digital unit cells in the PL. For most applications, this will be more unit cells than the amount of physical qubits which can be addressed with the eight DAC channels the chip offers. We performed various experiments using a superconducting five qubit chip to verify correct operation of the system. Future work will focus on performing more complex experiments and multi-system synchronization for further scaling.

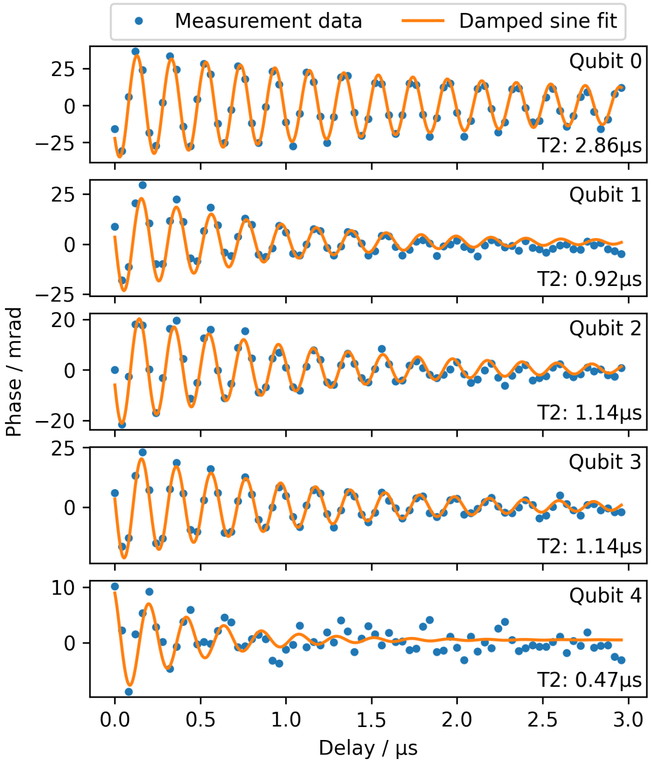


Fig. 8. Simultaneous measurements to extract the decoherence time T_2 with five qubits using a Ramsey pulse sequence. $100\,000$ repetitions have been performed for averaging.

Acknowledgment

Funding was provided by the Helmholtz Association. The authors acknowledge the financial support by the German Federal Ministry of Education and Research in the framework of PtQube (FKZ:13N15015). Richard Gebauer acknowledges support by the State Graduate Sponsorship Program (LGF). Nick Karcher acknowledges support by the Karlsruhe School of Elementary Particle and Astroparticle Physics (KSETA). We are grateful for the experimental support and infrastructure of the Institute of Physics at Karlsruhe Institute of Technology (KIT). We acknowledge Qkit [23] for providing a convenient measurement software framework for applications in quantum computing with superconducting quantum bits.

References

[1] E. Gerjuoy, "Shor's factoring algorithm and modern cryptography. an illustration of the capabilities inherent in quantum computers," *American Journal of Physics*, vol. 73, no. 6, pp. 521–540, 2005. [Online]. Available: https://doi.org/10.1119/1.1891170

[2] R. P. Feynman, "Simulating Physics with Computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, Jun. 1982.

[3] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, "Finding low-energy conformations of lattice protein models by quantum annealing," *Scientific Reports*, vol. 2, no. 1, p. 571, Aug. 2012. [Online]. Available: https://doi.org/10.1038/srep00571

[4] B. Bauer, S. Bravyi, M. Motta, and G. Kin-Lic Chan, "Quantum algorithms for quantum chemistry and quantum materials science," *Chemical Reviews*, vol. 120, no. 22, pp. 12 685–12 717, Nov. 2020. [Online]. Available: https://doi.org/10.1021/acs.chemrev.9b00829

[5] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress," *Reports on Progress in Physics*, vol. 81, no. 7, p. 074001, Jun. 2018. [Online]. Available: https://doi.org/10.1088/1361-6633/aab406

[6] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. L. Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, E. Solano, H. Neven, and J. M. Martinis, "Digitized adiabatic quantum computing with a superconducting circuit," *Nature*, vol. 534, no. 7606, pp. 222–226, Jun. 2016. [Online]. Available: https://doi.org/10.1038/nature17658

[7] B. Schumacher, "Quantum coding," *Phys. Rev. A*, vol. 51, pp. 2738–2747, Apr. 1995.

[8] 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, "Charge-insensitive qubit design derived from the cooper pair box," *Phys. Rev. A*, vol. 76, p. 042319, Oct. 2007.

[9] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, vol. 6, no. 2, p. 021318, 2019. [Online]. Available: https://doi.org/10.1063/1.5089550

[10] D. Ristè, C. C. Bultink, K. W. Lehnert, and L. DiCarlo, "Feedback control of a solid-state qubit using high-fidelity projective measurement," *Phys. Rev. Lett.*, vol. 109, p. 240502, Dec 2012. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.109.240502

[11] R. Gebauer, N. Karcher, D. Gusenkova, M. Spiecker, L. Grünhaupt, I. Takmakov, P. Winkel, L. Planat, N. Roch, W. Wernsdorfer, A. V. Ustinov, M. Weber, M. Weides, I. M. Pop, O. Sander, A. Fedorov, and A. Rubtsov, "State preparation of a fluxonium qubit with feedback from a custom fpga-based platform," *AIP Conference Proceedings*, vol. 2241, no. 1, p. 020015, 2020. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/5.0011721

[12] 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. 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. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, 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, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019. [Online]. Available: https://doi.org/10.1038/s41586-019-1666-5

[13] D. Rosenberg, D. Kim, R. Das, D. Yost, S. Gustavsson, D. Hover, P. Krantz, A. Melville, L. Racz, G. O. Samach, S. J. Weber, F. Yan, J. L. Yoder, A. J. Kerman, and W. D. Oliver, "3d integrated superconducting qubits," *npj Quantum Information*, vol. 3, no. 1, p. 42, Oct 2017. [Online]. Available: https://doi.org/10.1038/s41534-017-0044-0

[14] A. P. M. Place, L. V. H. 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, "New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds," *Nature Communications*, vol. 12, no. 1, p. 1779, Mar 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-22030-5

[15] A. Somoroff, Q. Ficheux, R. A. Mencia, H. Xiong, R. V. Kuzmin, and V. E. Manucharyan, "Millisecond coherence in a superconducting qubit," 2021.

[16] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. Girvin, L. Jiang *et al.*, "Extending the lifetime of a quantum bit with error correction in superconducting circuits," *Nature*, vol. 536, no. 7617, pp. 441–445, 2016.

[17] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, J. Heinsoo, J.-C. Besse, M. Gabureac, A. Wallraff, and C. Eichler, "Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits," *npj Quantum Information*, vol. 5, no. 1, pp. 1–7, 2019. [Online]. Available: https://doi.org/10.1038/s41534-019-0185-4

[18] Quantum Machines. (2019) The quantum orchestration platform. [Online]. Available: https://www.quantum-machines.co/platform/

[19] Zurich Instruments. (2019) Quantum Computing Control System. [Online]. Available: https://www.zhinst.com/others/quantum-computing-control-system-qccs

[20] Keysight. (2019) Quantum Engineering Toolkit (QET) - data sheet. [Online]. Available: https://www.keysight.com/us/en/assets/7018-06423/data-sheets/5992-3503.pdf

[21] N. Karcher, R. Gebauer, R. Bauknecht, R. Illichmann, and O. Sander, "Versatile configuration and control framework for real time data acquisition systems," *IEEE Transactions on Nuclear Science*, pp. 1–1, 2021.

[22] Cloud Native Computing Foundation. (2020) gRPC - a high-performance, open source universal RPC framework. [Online]. Available: https://grpc.io/

[23] Qkitgroup. (2020) Qkit - a quantum measurement suite in python. [Online]. Available: https://github.com/qkitgroup/qkit

[24] R. Gebauer, N. Karcher, J. Hurst, M. Weber, and O. Sander, "Taskrunner: A flexible framework optimized for low latency quantum computing experiments," in *2021 IEEE 34th International System-on-Chip Conference (SOCC)*, 2021.

[25] *Wishbone B4 - WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores*, OpenCores, 2010. [Online]. Available: https://cdn.opencores.org/downloads/wbspec_b4.pdf

[26] M. Kristen, A. Schneider, A. Stehli, T. Wolz, S. Danilin, H. S. Ku, J. Long, X. Wu, R. Lake, D. P. Pappas, A. V. Ustinov, and M. Weides, "Amplitude and frequency sensing of microwave fields with a superconducting transmon qudit," *npj Quantum Information*, vol. 6, no. 1, p. 57, Jun 2020. [Online]. Available: https://doi.org/10.1038/s41534-020-00287-w