Received 26 January 2024; revised 16 May 2024; accepted 27 May 2024; date of publication 5 June 2024; date of current version 26 June 2024.

Digital Object Identifier 10.1109/TQE.2024.3409811

FASQuiC: Flexible Architecture for Scalable Spin Qubit Control

MATHIEU TOUBEIX[1,2] (Graduate Student Member, IEEE), ERIC GUTHMULLER[1], ADRIAN EVANS[1], ANTOINE FAURIE[1], AND TRISTAN MEUNIER[2]

[1]CEA, List, Université Grenoble Alpes, F-38000 Grenoble, France
[2]CNRS, Institut Néel, Université Grenoble Alpes, F-38000 Grenoble, France

Corresponding author: Mathieu Toubeix (e-mail: mathieu.toubeix@cea.fr).

ABSTRACT As scaling becomes a key issue for large-scale quantum computing, hardware control systems will become increasingly costly in resources. This article presents a compact direct digital synthesis architecture for signal generation adapted for spin qubits that is scalable in terms of waveform accuracy and the number of synchronized channels. The architecture can produce programmable combinations of ramps, frequency combs, and arbitrary waveform generation (AWG) at 5 GS/s, with a worst-case digital feedback latency of 76.8 ns. The field-programmable gate array (FPGA)-based system is highly configurable and takes advantage of bitstream switching to achieve the high flexibility required for scalable calibration. The architecture also provides GHz rate, multiplexed, in-phase and quadrature component, single-side band modulation for scalable reflectometry. This architecture has been validated in hardware on a Xilinx ZCU111 FPGA demonstrating the mixing of complex signals and the quality of the frequency comb generation for multiplexed control and measurement. The key benefits of this design are the increase of controllability of ramps at the digital-to-analog converter (DAC) frequency and the reduction in memory requirements by several orders of magnitude compared with existing AWG-based architectures. The hardware for a single channel is very compact, 2% of ZCU111 logic resources for one DAC lane in the default configuration, leaving significant circuit resources for integrated feedback, calibration, and quantum error correction.

INDEX TERMS Direct digital synthesis, field-programmable gate array (FPGA), large-scale quantum (LSQ) computing, quantum control, spin qubits.

I. INTRODUCTION

A. LARGE-SCALE QUANTUM COMPUTING

Quantum computers need error correction to achieve quantum advantage. They also require calibration of large sets of parameters for the correct operation of qubits. Large-scale quantum (LSQ) computing requires fast, scalable, and flexible feedback to implement quantum error correction (QEC) and calibration. QEC based on surface codes requires measurements, complex computations, and dynamic control of the qubits. QEC must operate on thousands of qubits and react within about a microsecond (typical gate operation time) to prevent decoherence [1], [2], [3], [4]. Typical decoherence times in spin qubits are around a few milliseconds [5], [6] but error correction needs to be much faster than this to leave time for computation between corrections. Therefore, the two main requirements for QEC are scalability and latency.

Calibration is also necessary to improve qubit and gate fidelity, which is the first step toward scalable quantum computers. It involves different types of feedback, such as tracking qubit frequencies and optimizing control pulses for implementing gates or reflectometry-based measurements. For instance, calibration protocols from Google Sycamore [7] need 36 h at initialization and 4 h per day for only 53 qubits. The two main requirements for calibration are scalability and flexibility.

A typical qubit control experiment (see Fig. 1) involves a network of field-programmable gate arrays (FPGAs) at room temperature each handling the feedback loop for a limited number of qubits. FPGAs measure qubits and update control signals within a few hundred nanoseconds to prevent decoherence and additionally reduce the time required for calibration. FPGAs are currently the ideal choice for qubit control as they can achieve very low feedback latency, while leaving the necessary flexibility for emerging quantum control and QEC schemes. In the future, critical parts may benefit from an application-specific integrated circuit (ASIC)

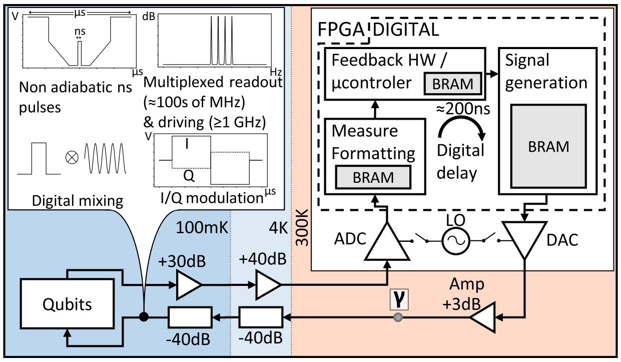


FIGURE 1. Architecture and typical waveforms for spin qubit control. γ is our measurement point for the experimental results (see Section III-C).

implementation, possibly at cryogenic temperatures [8], but FPGAs are ideal for early designs due to their flexibility.

Spin qubits require specific quantum control, which can be divided into two main categories. First, quasi-static biasing, specific to spin qubit control, is required to manipulate qubit state by changing the chemical potentials and tunnel barriers of quantum dots. They require fine temporal and amplitude control of ramps [9] while achieving high V/ns slopes for nonadiabatic quantum state transitions. Second, similar to superconducting qubits, driving pulses (modulated sine waves) are used both for qubit control (GHz range) or measurement (100 MHz range). Simultaneous generation of multiple driving pulses enables online multiplexing, which is an important feature for scaling quantum computing, especially for reflectometry-based measurements [10]. For the latter, in-phase and quadrature component (I/Q) modulation and demodulation based on the combination of an in-phase (I) and quadrature (Q) signal are required to generate precise single-side band (SSB) modulation and retrieve both signals at measurement time.

An example of quasi-static control is illustrated in Fig. 2 [11]. It shows the implementation of a SWAP gate with a double quantum dot and control of the detuning between the two dots gates. This kind of control can be used to implement two qubit gates for a one-electron qubit but is also used for a one qubit gate in the case of singlet–triplet qubits (two electrons qubits) or exchange only qubits (three electrons qubits) [12], [13], [14], [15]. The detuning represents the differential voltage of the gates of each dot. By changing this detuning abruptly (nonadiabatically) or slowly (adiabatically), we control whether the electrons transition when encountering crossed quantum states (Landau–Zener crossing) [16]. A first path is applied starting from a highly detuned singlet state $|S\rangle$ where both electrons are in the same dot. At A, the electrons wait a separation time τ_s, which should be long enough to reach state $|\uparrow\downarrow\rangle$. Typically this requires a few tens of nanoseconds. By driving the system abruptly with an exchange pulse at detuning A_{ex}, Rabi oscillations will occur making the system oscillate between $|\uparrow\downarrow\rangle$ and $|\downarrow\uparrow\rangle$. Waiting the right amount of exchange time τ_{ex} (typically from 0.1 to 3 ns), we can go to $|\downarrow\uparrow\rangle$ state, apply a reverse path, and attain the triplet state $|T_0\rangle$. A $\sqrt{\text{SWAP}}$ gate, which is a valid 2-qubit gate to constitute a universal set of quantum gates, can also be implemented by waiting half the exchange time. In this sequence, we wish to minimize τ_s to achieve fast gate implementation. τ_{ex} and A_{ex} must be finely tuned to achieve high gate fidelity. Fine control of the slopes is critical, and is one of the focuses of the proposed architecture. These are examples of parameters, which must be determined for each individual qubit, and that require fast calibration for an LSQ computer.

Digital mixing can be applied to achieve more complex control schemes and pulse engineering [17] to further increase gate fidelity by optimizing the shapes of the control signals. Digital signal generation is a way to increase the flexibility in an ever-changing experimental context and reduce analog noise sources.

B. STATE OF THE ART

Most architectures (see Fig. 3) in the state of the art focus on superconducting qubits and therefore have arbitrary waveform generations (AWGs) with high memory consumption and low flexibility to configure and change stored waveforms. These architectures are limited to switching between pregenerated waveforms for feedback [18], [19], [20], thus fast feedback is possible only for very simple schemes. With the increasing number of qubits and the increasing complexity of QEC schemes, the cost of storing pregenerated waveforms is also increasing. The time to generate these waveforms dominates the overall time required for calibration. COMPAQT [21] reduces the memory consumption of AWG with discrete cosine transform (DCT) and run-length encoding (RLE) compression, providing gains on quasi-static control, but still lacks fast hardware feedback capabilities (since reconfiguration of signals requires a software step) and lacks digital mixing. The calibration phase is also slowed by the compression of stored waveforms. Single channel, on-the-fly generation of sine waves has been achieved up to dual-tone generation [22], [23]. Presto [24] does achieve multitone generation, but this is done by combining physical ports inside the digital-to-analog converters (DACs) on the FPGA board, resulting in a loss of scalability due to the reserved channels. On-the-fly generation of ramps [25] is currently only done at the internal FPGA clock frequency therefore limiting their temporal controllability. None of these architectures exploit the configurability of FPGAs to facilitate the calibration phase. These other qubit control architectures lack flexibility and the specific features for controlling spin qubits.

C. CONTRIBUTION

We propose a flexible architecture for scalable spin qubit control (FASQuiC) using on-the-fly generation of ramps and frequency combs to reduce the memory requirements and to enable subnanosecond scale dynamic signal synthesis. The architecture is optimized for the scalability and flexibility of

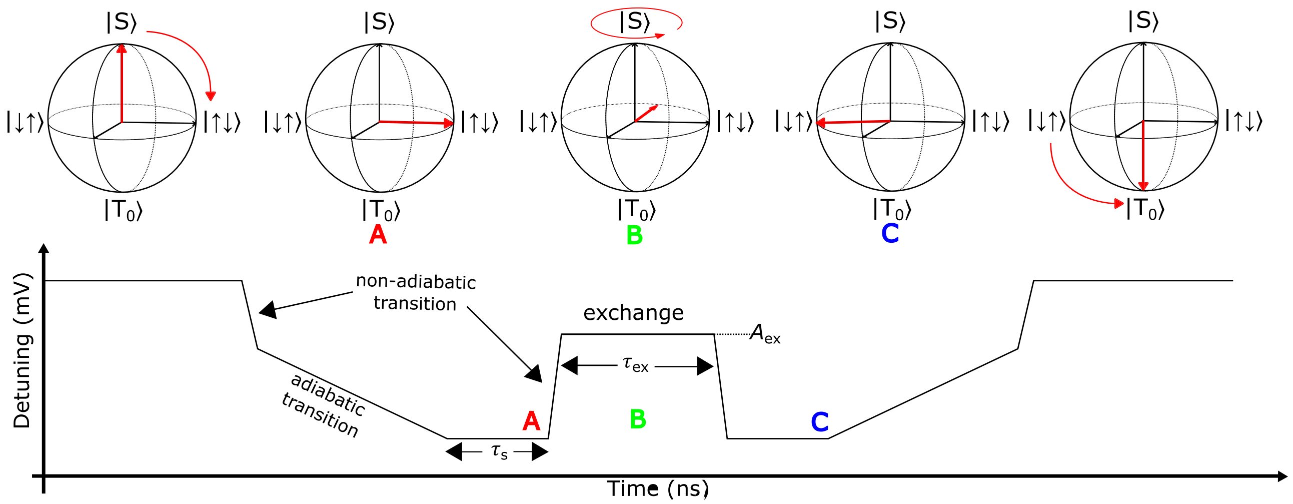
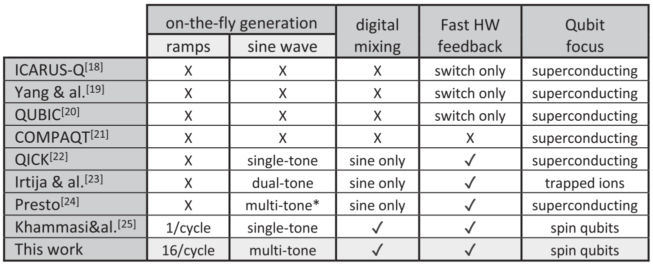


FIGURE 2. Example of a quasi-static control of a double quantum dot. |S⟩ and $|T_0\rangle$ states encode either the 1 or the 0 states for a 2-electron qubit or a 2-qubit state for a one electron qubit and are represented on the Bloch sphere. Detuning is the differential gate voltage between quantum dots and is controlled by room temperature hardware and an attenuation chain to the refrigerator.



	on-the-fly generation		digital	Fast HW	Qubit
	ramps	sine wave	mixing	feedback	focus
ICARUS-Q[18]	X	X	X	switch only	superconducting
Yang & al.[19]	X	X	X	switch only	superconducting
QUBIC[20]	X	X	X	switch only	superconducting
COMPAQT[21]	X	X	X	X	superconducting
QICK[22]	X	single-tone	sine only	✓	superconducting
Irtija & al.[23]	X	dual-tone	sine only	✓	trapped ions
Presto[24]	X	multi-tone*	sine only	✓	superconducting
Khammasi&al.[25]	1/cycle	single-tone	✓	✓	spin qubits
This work	16/cycle	multi-tone	✓	✓	spin qubits

FIGURE 3. State of the art of control architectures for qubit manipulation. The key benefits of this work are the increase in controllability of ramps, flexible on-the-fly generation, and improved scalability compared with other architectures.

quasi-static and driven control of semiconductor spin qubits in a cryogenic environment. FASQuiC, with its flexibility, can also meet the current experimental needs of other types of qubits, such as superconducting qubits, regardless of the final application or architecture.

This approach alleviates the need for high bandwidth for waveform transfer, which is a major overhead of existing architectures. Our primary contribution is a novel ramp generator, which generates ramps at the oversampled DAC frequency to achieve temporal controllability of the ramps down to 200 ps (5 GS/s) and improve quasi-static control of spin qubits. The initial signal generation architecture proposed in [26] has been extended and characterized.

FASQuiC can mix several sources between a ramp generator, a sine wave generator and an AWG to create complex signals. The sine wave generator can create up to 16 frequencies using one DAC lane.

FASQuiC supports I/Q modulation and demodulation in SSB mode enabling scalable multiplexed reflectometry by generation of frequency combs in the first and second Nyquist zones. FASQuiC has been validated in hardware on a ZCU111 FPGA board from Xilinx, for which we have characterized the phase noise (PN) of the oscillator and embedded phase-locked loop (PLL). FASQuiC is a highly parameterized design, which allows it to fully exploit the reconfigurability of FPGAs. Therefore, it can expand its scope of action enabling scalable calibration.

II. PROPOSED ARCHITECTURE

A. SIGNAL GENERATION

1) COMPLEX SIGNAL GENERATOR

The complex signal generator (CSG) is responsible for generating the digital signals driven to the DACs. The microarchitecture of one CSG is shown in Fig. 4. Each CSG outputs N_{out} points per hardware clock cycle at 312.5 MHz, ($1/N_{\text{out}}$ of the DAC frequency) to feed the 14-bit DAC working at the higher DAC clock frequency to achieve oversampling. In our design, N_{out} is chosen at compilation time (from 1 to 16 in steps of powers of two).

A CSG can be configured, at compilation time, to embed between one and three unique generators that will be mixed together to create more complex control schemes. Each generator can either be a ramp generator, a sine comb generator, or an AWG.

The merger block can mix the outputs of the two first generators. Dynamically, it can act as a multiplexer, add the two signals with saturating arithmetic, or normalize and multiply the signals. If the CSG has only one generator, this block delays the signal by two cycles to match the latency of other CSGs in the system.

The modulator block can dynamically modulate the output of the merger with a frequency comb or leave it unchanged. This frequency comb can be a sine comb from the third generator or a cosine comb from another CSG, this feature is chosen at compilation time. As with the merger, this block delays the signal by two cycles if it is not needed.

Generators have a common structure. Block random-access memories (BRAMs) or registers are used to store waveform parameters, those parameters are written or read

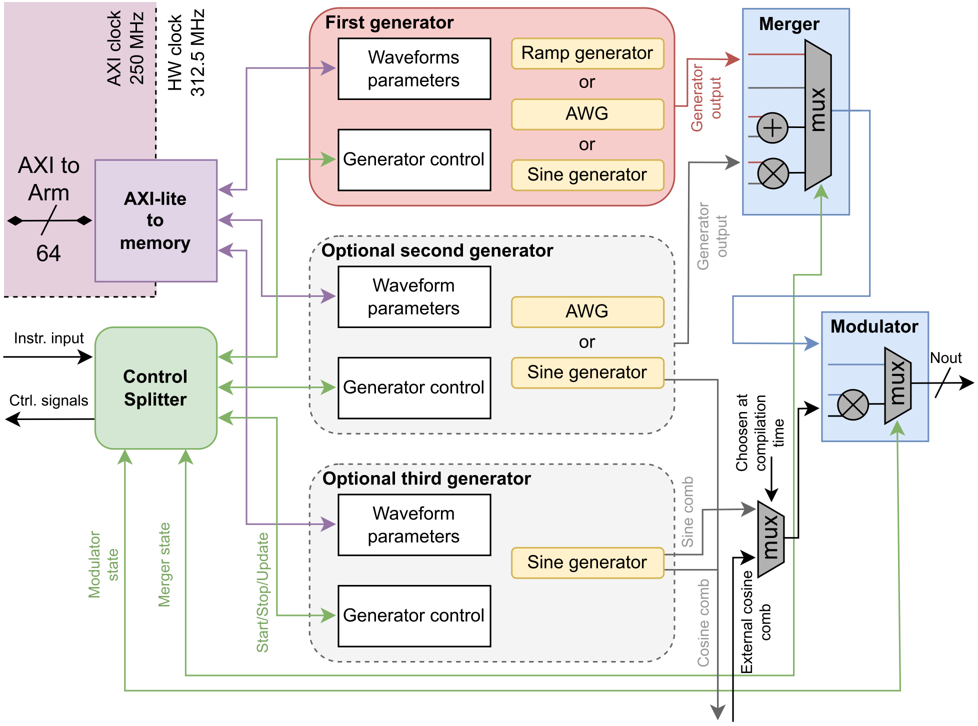


FIGURE 4. Architecture of one CSG.

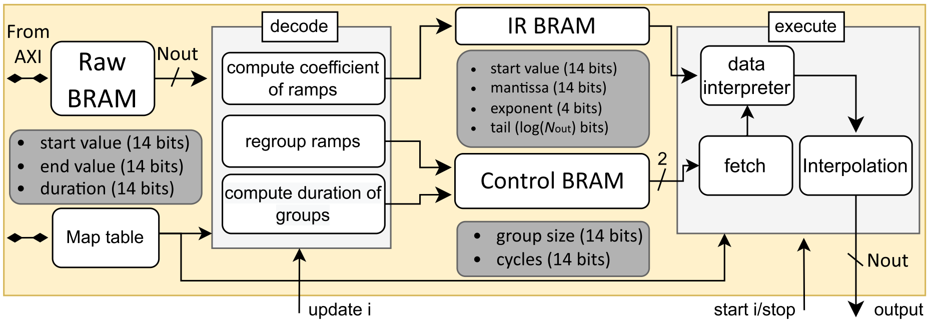


FIGURE 5. Architecture of the ramp generator.

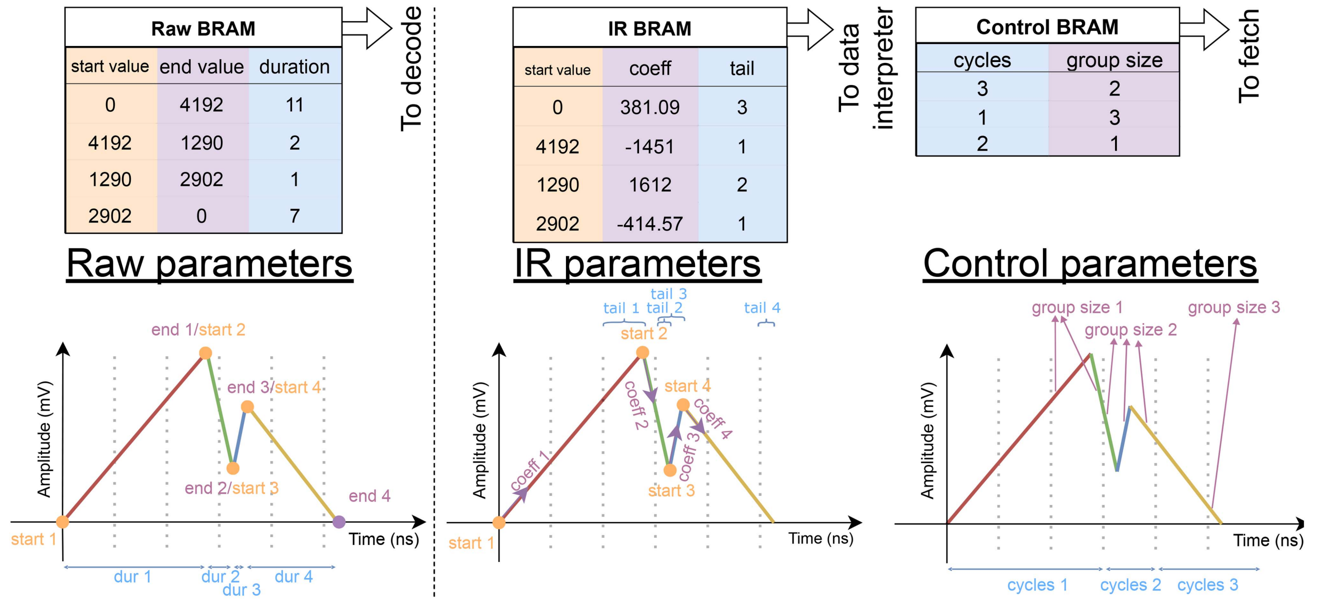
back through an advanced extensible interface (AXI) network (Section II-C2). Each generator is manipulated via control signals sent by a controller block (Section II-A5). This controller can also dynamically change the configurations of the merger and the modulator. Finally, the controller is aware of the state of each generator and optionally sends interrupts to the control status registers (CSRs) (Section II-C).

2) RAMP GENERATOR

The ramp generator (see Fig. 5) generates all waveforms, which can be decomposed as successions of linear parts. It allows temporal controllability at $1/N_{\text{out}}$ of a hardware clock cycle, required for fine control of nonadiabatic pulses. An example of such control is shown in Fig. 12(a).

Ramp waveform parameters are stored in the raw BRAM. For each ramp waveform, its ramps are described by their starting value, ending value, and duration in DAC clock cycles. The ramp waveforms are referenced in a map table, via a header word describing the location of the ramp waveform parameters in the BRAMs of the ramp generator and the ramp waveform's length. With this indirection, the controller can access several ramp waveforms with different sizes stored in the ramp generator BRAMs and enable scalable complex hardware feedback. A ramp waveform can run in a continuous mode by setting a bit in its header word, thus reducing storage for repetitive patterns; for example, waveform ③ in Fig. 14 requires only three words to store its parameters and header.

Existing algorithms [27] only interpolate one ramp per hardware clock cycle. Using brute-force parallelism would require 16 radix-16 blocks to satisfy a 5 GS/s generation, which is overly costly in logic resources. The design must be able to dynamically generate from one to 16 ramps in one



Raw BRAM		
start value	end value	duration
0	4192	11
4192	1290	2
1290	2902	1
2902	0	7

IR BRAM		
start value	coeff	tail
0	381.09	3
4192	-1451	1
1290	1612	2
2902	-414.57	1

Control BRAM	
cycles	group size
3	2
1	3
2	1

FIGURE 6. Example of control path and IR computation, the dashed lines represent the hardware clock cycles. On the left-hand side, the raw parameters describe all ramps by their starting value, ending value, and duration in DAC clock cycles. After decode, those ramps are transformed into an IR describing their starting value, their slope, and their relative DAC clock cycle after their last hardware clock cycle. They are also merged into groups of ramps in the control BRAM containing the size of their group and the number of hardware clock cycle they last.

hardware clock cycle without interrupting the pipeline, therefore at each cycle, the generator must be aware of the number of ramps it has to compute for the next cycle. To achieve this level of controllability, ramps are first transformed from raw parameters into an intermediate representation (IR), this process is presented in Fig. 6. For each ramp, the IR describes the starting value and the coefficient of the ramps encoded with a floating point representation. This coefficient is necessary to enable interpolation for a dynamic fraction of a hardware clock cycle.

The duration parameters are then used to regroup ramps that finish in the same hardware clock cycle. Merging ramps in groups that last at least one cycle ensures that the execute stage always knows the next ramps to play. Otherwise, the execute stage would need to fetch and analyze the maximum number of ramps at each cycle, which would be more costly in resources than our approach.

Each group of ramps is described by its duration in hardware clock cycles (*cycles* column) and the number of ramps it contains (*group size* column).

We also extract the DAC clock cycle at which the ramp finishes inside its last hardware clock cycle (*tail* column). This *tail* parameter is stored in the IR and is used by the data interpreter to organize the ramps inside one hardware clock cycle when several ramps are played at the end of a group of ramps.

After the decoding, ramp duration parameters have been decomposed into: 1) the information required on a hardware clock cycle basis (*group size* and *cycles*), directly usable by the execute stage and 2) DAC clock cycle information (*tail*) for completing the last hardware clock cycle of the group and switching to the next group of ramps.

After the first IR word is written into: the IR BRAM, the waveform is ready to run in the execute stage, even if the decoding has not finished yet. The execute stage fetches the control parameters to analyze the group of ramps. The data interpreter then starts to play the first ramp of the group. At each hardware clock cycle, the starting point of the first ramp is incremented by the coefficient shifted $\log N_{\text{out}}$ times, a simple interpolation at the output of the data interpreter computes the oversampled points. When a group of ramps finishes, even after less than one cycle, the data interpreter completes the hardware clock cycle with the rest of the ramps in the group and the execute stage fetches the next group. The interpolator then uses the starting points and coefficients of each ramp played during the hardware clock cycle to generate the N_{out} points given to the DAC. The worst-case error from summing slope coefficients to compute ramps is ±1 bit out of 14 compared with a perfect ramp ($\pm 31\ \mu$V for a 500-mV full scale).

Once the IR is computed, it is stored and can be reused, enabling fast switching between two already updated ramp waveforms. The process is pipelined to ensure N_{out} points are generated every hardware clock cycle. The hardware maintains the last known value when a ramp waveform finishes playing. This approach prevents glitches, protecting cryogenic samples from undefined behaviors.

3) SINE WAVE GENERATOR

The sine wave generator (see Fig. 7) produces sine combs and cosine combs for multiplexed modulation (Section II-B).

The existing single sine generator IP (Xilinx PG141 [28]) is based on lookup tables and a Taylor expansion for refinement. In our architecture, N_{out} of these are instantiated

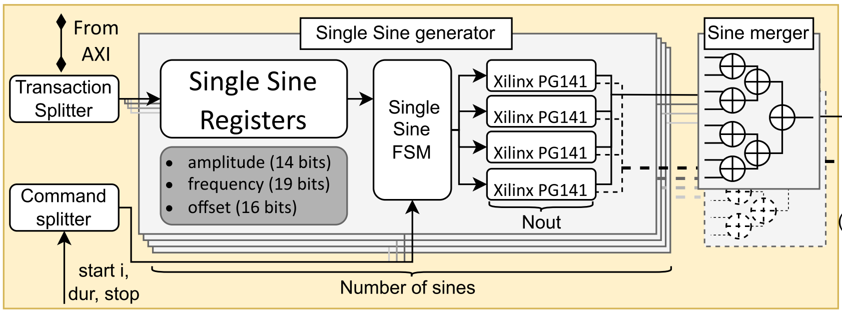


FIGURE 7. Architecture of the sine wave generator.

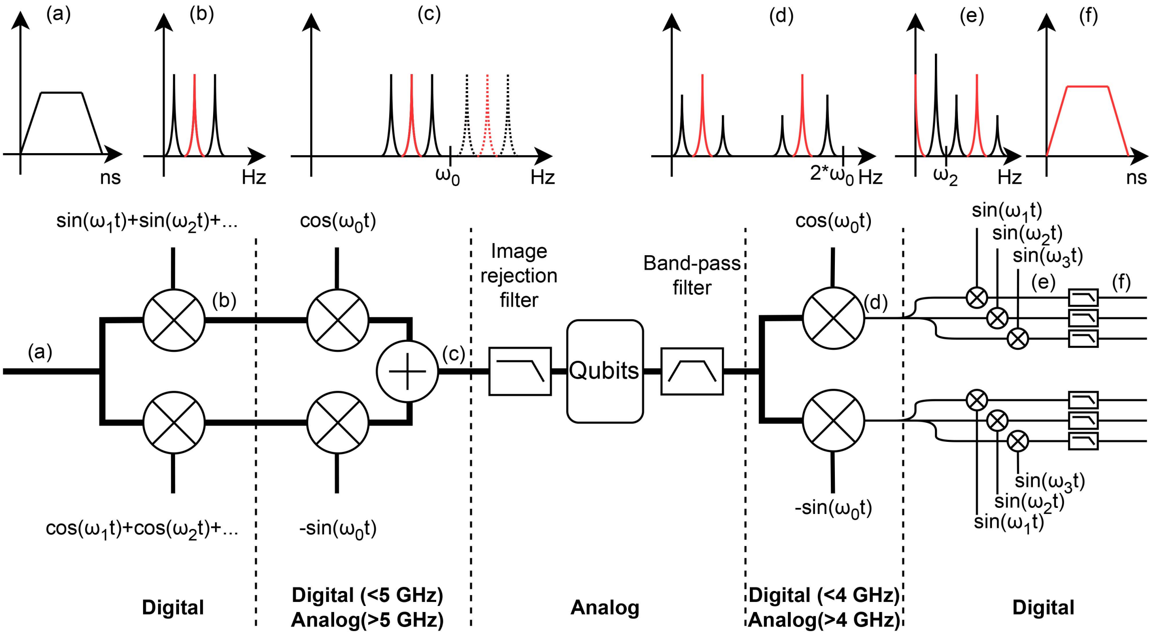


FIGURE 8. Example of a multiplexed reflectometry setup. A time-domain signal (a) is first multiplied by an I and Q frequency comb (b). The two pulses are then upconverted (c) to achieve I/Q modulation for driving control pulses or reflectometry measurements. In our design, this upconversion can either be digital or analog. After going through the qubits, for measurement, the signal is downconverted (d), demultiplexed (e) and averaged (f) to retrieve all reflected information.

in parallel, configured at $1/N_{\text{out}}$th of the desired frequency and shifted accordingly in phase, to create oversampling. Multiple, oversampled sine waves are combined to create a frequency comb.

The generator can also produce an optional cosine comb alongside the sine comb to allow digital SSB generation for I/Q modulation. This feature is selected at compilation time.

The maximum number of sine waveforms generated by the generator is chosen at compilation time. As merging sine waves is done by averaging them, this maximal number of sine waves defines the maximal amplitude of each single sine and must be chosen carefully. Waveform parameters for sine waveforms are stored in registers instead of BRAMs since the maximal number of words to store is small (16).

4) AWG

The AWG can generate arbitrary waveforms stored point by point in its data BRAM. For every hardware clock cycle, it fetches N_{out} points in the data BRAM to achieve oversampling.

This standard AWG block is augmented with the ability to repeat a point up to 2^{14} hardware clock cycles to reduce the memory footprint. A flag in the data BRAM word triggers this repeat mode. Note, the repeat command does not need to be aligned with the hardware clock.

Similarly to the ramp generator, a map table allows the controller to indirectly access the different waveforms stored in the data BRAM. A waveform can play continuously by turning on a bit in its header word.

5) CSG CONTROLLER

The architecture is fully distributed: each CSG is connected to an external controller, as shown in Fig. 9, using custom instructions composed of control instructions and flow management instructions. Some of these instructions are shown in Fig. 12. The advantage of a fully distributed system for

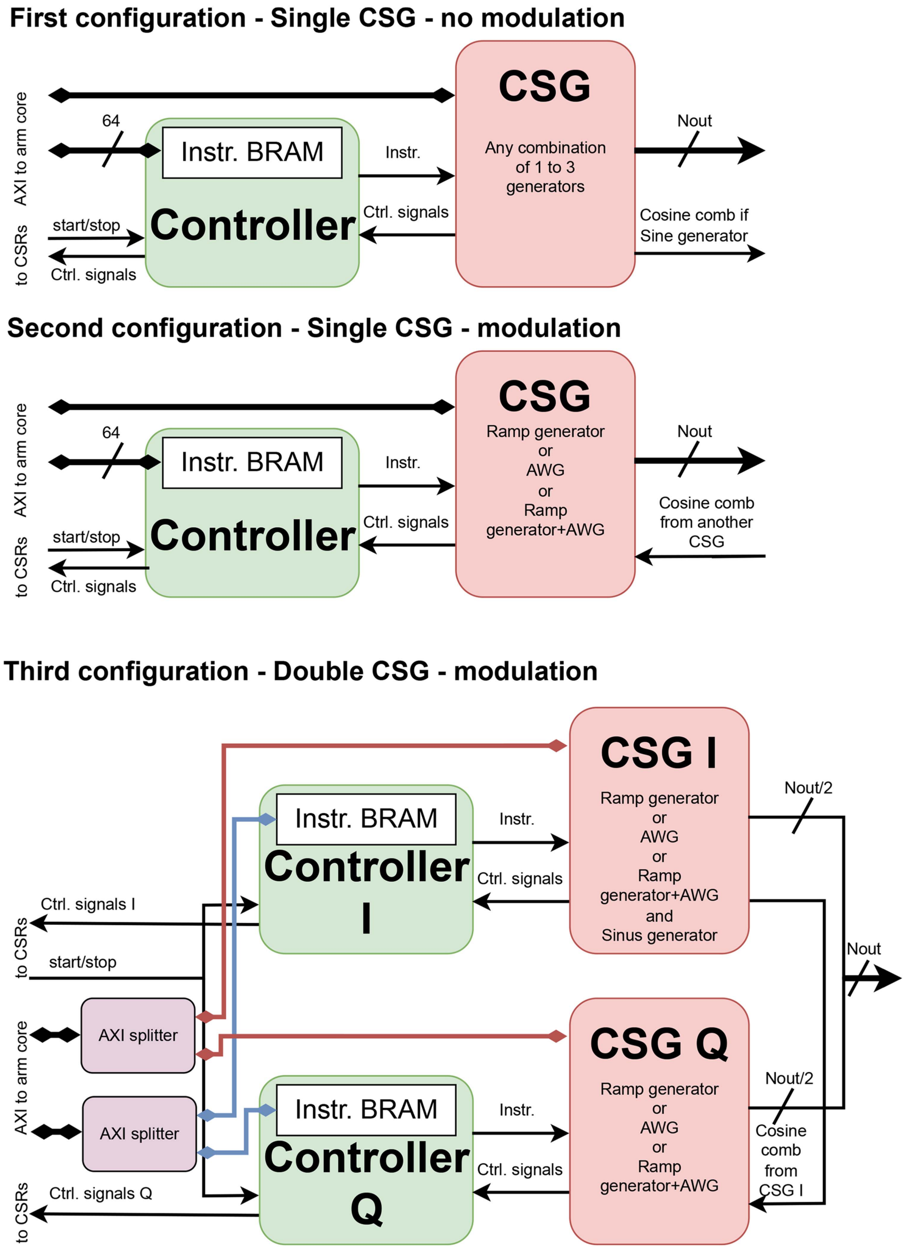


FIGURE 9. Three CSG configurations cover all needs from quasi-static control to multiplexed I/Q modulation.

control is the ability to implement quantum feedback on each CSG. As those controllers are fully deterministic, they can be synchronized to interpret centralized global feedback results and take local decisions.

Control instructions include all instructions to communicate with the generators. For the ramp generator, ramp waveforms, addressed by their index in the map table, can be updated, started, or stopped. Waveforms from the AWG, addressed by their index in the map table, can be started or stopped. Finally, sine wave generation can be started or stopped; the instruction specifies the number of sines to start and the duration in hardware clock cycles it has to be played. A zero duration will run the sine comb continuously.

Flow management instructions include wait statements to freeze the controller either for a fixed duration, until an event from a synchronization protocol (not presented here) or for a change in the state of a generator. For example, if a CSG embeds a ramp generator, the controller can wait for the first word of a ramp waveform update to be written in the BRAM, then it can chain with starting this ramp waveform without dead cycles. Flow management instructions also include loop capabilities, up to 16 nested loops can be run.

The instructions are stored in a dedicated BRAM for each controller, accessible through an AXI network. Controllers can be started or stopped at any time. They also share generator states and their own state to the CSR block and generate interrupts when an unexpected behavior occurs.

B. I/Q MODULATION AND DEMODULATION

1) BACKGROUND ON I/Q MODULATION

I/Q modulation is a quadrature amplitude modulation that can be used both for driving and measurement. For driving, it allows fine tuning of input signal phase and enables pulse engineering where the shapes of the I and Q spectra are optimized to increase the fidelity of gates. For measurement, it is necessary for reflectometry-based methods [29], [30], [31], [32]. Reflectometry measurement senses a change of capacitance, called quantum capacitance, for example of a double quantum dot due to its quantum state. Reflectometry is a good candidate for LSQ computers since it can be multiplexed, tends to be faster than measurements based on transport with single electron transistor. Some techniques, such as gate reflectometry, also reduce device footprint for implementing the measurements.

An example of a multiplexed reflectometry measurement achievable with the ZCU111 is shown in Fig. 8. In terms of signal generation, to achieve SSB modulation and increase measurement sensitivity, a window is modulated by an I and a Q low-frequency comb in Fig. 8(a) and (b), respectively. This step enables SSB modulation, without it the upper band [dashed line in Fig. 8(c)] would appear in the next step. The two resulting combs can then be upconverted and modulated by an I and Q carrier frequency resulting in one, high-frequency comb. This carrier frequency can be analog and external to the FPGA board for very high frequencies or internal and digital for lower frequencies. Second Nyquist zone techniques [33] help achieve higher frequencies with digital modulation.

Only the lower frequency comb needs to be updated to achieve fast feedback. An image rejection filter can be added at the output of the DACs to avoid coupling to DAC replicas in second Nyquist zone when demodulating.

In terms of measurement, the reflected signal is first filtered to reduce thermal noise at the input of the analog-to-digital converter (ADC) with a passive bandpass filter. Then, the signal is downconverted by the same upconversion carrier frequency in Fig. 8(d). Finally, each frequency of the lower frequency comb is demodulated in Fig. 8(e) and a low-pass filter in Fig. 8(f) retrieves the full quadrature information of the symbol. A discriminator then splits the measurements of zeros and ones in the I/Q plane to finally measure the quantum state.

2) I/Q MODULATION

In order to use the CSG for I/Q modulation three different configurations are possible, as shown in Fig. 9. The first configuration is the default and is used for quasi-static control of qubits. One controller is directly connected to a CSG with any combination of generators. For multiplexed I/Q modulation, this configuration represents the I part of the signal for an external carrier frequency.

In the second configuration, a cosine comb can be connected from another CSG in the first configuration to generate the Q part of the signal. The two signals can then be upconverted outside the FPGA board.

The third configuration represents full digital multiplexed I/Q modulation. Two CSGs are connected to the same DAC of the Xilinx ZCU111. Each one works at half the oversampling rate of the design, their points are then interleaved to feed the DAC IP that will apply the I/Q modulation. This configuration is also useful for future pulse engineering experiments since both spectra of the quadrature can be manipulated in parallel.

3) I/Q DEMODULATION

Hardware for multiplexed I/Q demodulation is shown in Fig. 8(e) and (f). Demodulation is achieved by using the single sines generated by the CSG. After demodulation each frequency spectrum is filtered through a low-pass filter to retrieve the I/Q amplitudes of the reflected signal. This low-pass filter is implemented as follows: first, the signal is downsampled from the ADCs' clock to the hardware clock at 312.5 MHz. Then, moving averages are applied, the size of these moving averages can be dynamically configured before starting the measurement. The advantage of a moving average is its low cost in resources (one BRAM and two adders) compared with an FIR or IIR filter while it is sufficient to retrieve the low-frequency spectrum of each quadrature. The results can be directly distributed to future hardware feedback blocks either before or after demodulation and filtering. Otherwise, they are stored in RAM buffers accessible from the processor through an AXI network for software post-processing.

C. SYSTEM ARCHITECTURE

The top-level design implemented on the ZCU111 is shown in Fig. 10. The embedded advanced RISC machine (ARM) processor runs an application that communicate with the different hardware blocks through two AXI networks.

1) CSRS

The CSR block has two responsibilities. First, it relays the commands sent by the ARM processor to the hardware in a synchronized manner. All DAC lanes are started or stopped at the same time. Then, the CSRs also gather all the status signals from the generators and controllers, and detect anomalies, such as an AXI error or an unsupported command in a controller. If an anomaly is detected, an interrupt is sent to the software part, and mechanical relays at the output of the DACs are opened to protect the samples. Finally, the hardware design configuration is stored in read-only CSRs facilitating the software when switching bitstreams.

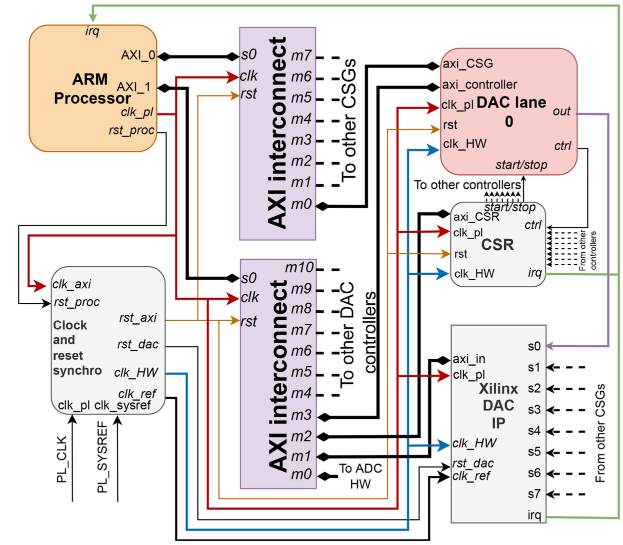


FIGURE 10. Top-level architecture implemented on the ZCU111.

2) AXI NETWORK, CLOCKING AND MEMORY MAPPING

Two AXI networks connect the ARM processor to the signal generation hardware to reduce congestion. The first one gives the ARM core access to the CSGs' BRAMs to initialize all waveform parameters. Each CSG has its own page, which is divided into subpages at compilation time depending on the CSG configuration. The other network is used to access the controllers' instruction BRAMs, the CSR registers, the stored measurements, and Xilinx DAC IP configuration registers.

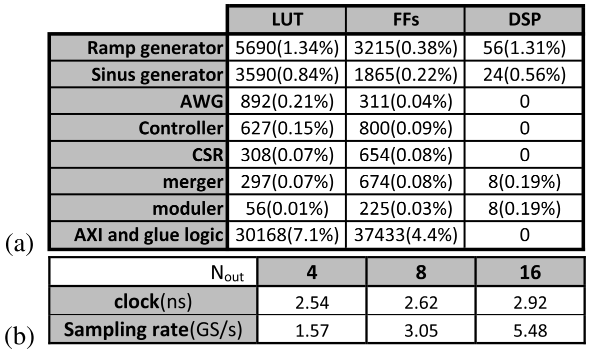
On our card, the internal oscillator has been changed to a 20-ppm 125-MHz oscillator (Crystek Cvhd-950-125.000), which generates a 125-MHz clock PL _ CLK that is upconverted to the hardware clock at 312.5 MHz. The same PLL also generates a 2.5-MHz PL _ SYSREF clock from an external primary 10 MHz clock to synchronize the DACs' outputs across the different tiles of the DACs. This 10 MHz clock is distributed across multiple FPGAs to achieve a multiboard synchronized system.

3) SOFTWARE

The software runs on the embedded environment of the ZCU111 [34], [35], composed of an ARM processor, a Petalinux distribution, and a C library for hardware control provided by Xilinx. The program performs read and write operations to the BRAMs and registers of the system.

The program can also activate most features of the Xilinx DAC IP (digital upconversion for I/Q modulation, Multitile synchronization of DACs and ADCs, filtering, modulation correction, and sampling rate tuning).

In addition to the embedded software, another application runs on a distant host machine, which is in charge of compiling the parameters of the waveforms. These waveforms are described by the user in dedicated files. The output is subsequently fed to the software running on the ZCU111.



(a)

	LUT	FFs	DSP
Ramp generator	5690(1.34%)	3215(0.38%)	56(1.31%)
Sinus generator	3590(0.84%)	1865(0.22%)	24(0.56%)
AWG	892(0.21%)	311(0.04%)	0
Controller	627(0.15%)	800(0.09%)	0
CSR	308(0.07%)	654(0.08%)	0
merger	297(0.07%)	674(0.08%)	8(0.19%)
moduler	56(0.01%)	225(0.03%)	8(0.19%)
AXI and glue logic	30168(7.1%)	37433(4.4%)	0

(b)

N_{out}	4	8	16
clock(ns)	2.54	2.62	2.92
Sampling rate(GS/s)	1.57	3.05	5.48

FIGURE 11. (a) FPGA resource usage (with percentage of total ZCU111 resources) for $N_{out} = 8$, and AXI and glue logic cost is for eight DACs activated. BRAM cost is completely dependent on user choices since all BRAM sizes are configurable. (b) Timing for worst-case configuration with one sine wave and 8 kB BRAMs.

In addition, interrupt detection and security features are also implemented to halt the execution whenever an unforeseen behaviour is detected.

4) DESIGN CONFIGURATION

The SystemVerilog register transfer level (RTL) code is fully parameterized from a single parameter file, which configures the design. From a functionality point of view, the oversampling factor N_{out}, number of activated DACs, number of sines generated for a sine generator, and the type of internal generators for each CSG can be tailored to the experimental needs of each channel.

From a resource cost point of view, all the BRAM sizes can be changed up to 32 KB before timing requirements imply additional buffers in the design. Because indirection is used in the generators, the data did not have to be aligned based on the oversampling factor, which means that the storage can be fully utilized.

The bitstream can be dynamically updated by the ARM processor, which eases switching the hardware configuration for calibration. Applying the first bitstream takes 228 ms on average but this is reduced to 110 ms by using partial configuration for the subsequent bitstreams. For example, tracking qubit frequencies [36] slowly drifting over minutes would need a hardware configuration with heavy computational power for Bayesian estimation. Once the qubit frequency is found, the design could switch back to another configuration for standard control of the device [37]. In this way, the system can quickly adapt to the computational needs of each step during calibration while using a limited number of resources at each step.

5) RESOURCES AND TIMING

FPGA resource usage and timing results are presented in Fig. 11. For the sine generator, the presented cost is for a single sine wave. This must be multiplied by the number of sine waves (up to 16 frequencies) to get the full resource estimation. Above 16 frequencies, the hardware costs grow

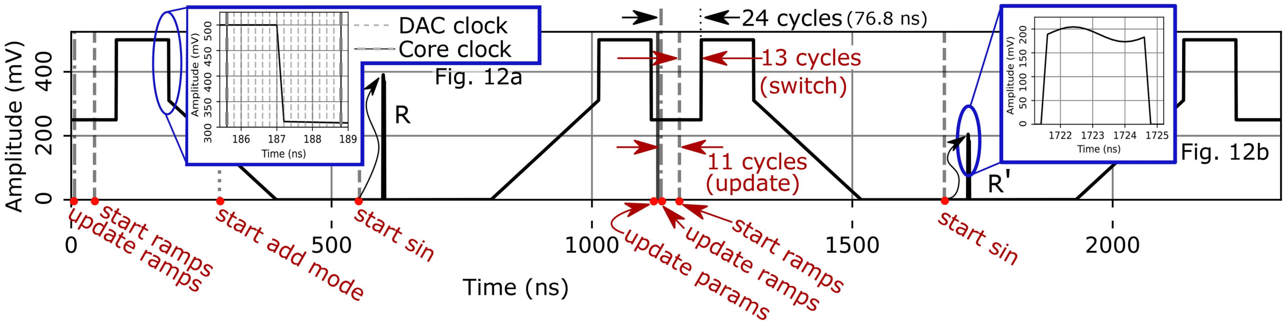


FIGURE 12. Hardware simulation of three CSG lanes with $N_{\text{out}} = 16$, one ramp generator and one sine wave generator. (a) Oversampled controllability of ramps. (b) Synchronization of a sine wave and ramp generators.

nonlinearly as additional buffers are needed to meet timing. A single DAC channel in the standard configuration uses less than 2% of the logic resources of the board, leaving space to implement hardware feedback. The BRAM cost is completely dependent on user choices since all BRAM sizes are configurable to fit experimental needs.

The design can generate DAC data at up to 5.48 GS/s. The ZCU111 internal carrier frequency generation, with the second Nyquist zone [33], can reach 5 GHz modulation. The critical timing path resides in the ramp execute stage (unless one of the BRAMs in the ramp generator exceeds 16 kB).

III. RESULTS

In this section, we first present the results of digital simulations of one CSG and its controller in isolation. We then show results for I/Q modulation (perform by the CSG) and demodulation (Section II-B3). Our digital testbench, running on Questa 2021.4_2 (Siemens EDA), can simulate any design configuration and drives the design from the AXI outputs of the processor and the ADC outputs while monitoring the inputs to the DACs. We then discuss the scalability of our design compared with other systems presented in state of the art. Finally, we present lab measurements, which demonstrate that FASQuiC successfully generates all mixing combinations of generators and produces high-quality frequency combs for I/Q modulation.

A. HARDWARE SIMULATION

1) CSG SIMULATION

The digital simulation in Fig. 12 shows an example of a feedback sequence, which uses one CSG (one ramp, one sine generator), an oversampling factor of 16, and a hardware frequency of 312.5 MHz. The sequence updates the plateau of the ramp (R, R'). This is a typical parameter that needs to be calibrated.

All delays in FASQuiC are fixed and synchronization between generators is possible. For instance, Fig. 12(b) shows the synchronization of both generators by adding their signals during the exchange part of the control signal.

Fig. 12(a) shows the capacity of the ramp generator to oversample the control of ramp generation. While classical digital ramp generators are limited to interpolation, this design can control up to N_{out} ramps per cycle, generating multiple on-the-fly ramps in the middle of a hardware clock cycle.

The dashed lines show the different control signals sent by the controller to achieve this sequence. Between the two repetitions of the signal, the parameter of the ramp R is changed to a lower value in the RAW BRAM of the ramp generator. The ramp signal needs to be updated before restarting it. It takes 35.2 ns (11 cycles) to compute and store the first word of the IR, then 41.6 ns (13 cycles) are needed to restart the ramp generator. Therefore, a feedback sequence for a ramp generator takes 76.8 ns (24 cycles), although switching between two already updated ramp waveforms costs only 13 cycles.

The sine generator takes at most 48 ns (15 cycles) for full feedback and 44.8 ns (14 cycles) for fast switching. The AWG takes 19.2 ns (6 cycles) for fast switching. Thus, our system is ready for fast synchronized feedback.

2) I/Q MODULATION AND DEMODULATION SIMULATION

In this section, a hardware simulation of an I/Q modulation/demodulation is detailed (see Fig. 13). CSGs in the third configuration are initialized at $N_{\text{out}} = 16$, ramp generators, and a sine generator with four sines. The demodulator is configured with an averaging window of size 1024 (410 ns integration time compatible with fast spin qubits readout). In practice, this parameter depends on the noise level.

When the controllers are started, the ramp generators start playing a symbol [see Fig. 13(a)], which is a trapezoidal pulse. The ramp generator could be replaced by an AWG to create a Gaussian pulse also used in superconducting qubits or optimized pulse control. A combination of a ramp generator and an AWG can also be used for more complex pulse shaping. Another advantage of using ramp generators is the fast reconfigurability of the symbol in a shape optimization process, thus, any Gaussian pulse could be approximated by a succession of configurable ramps.

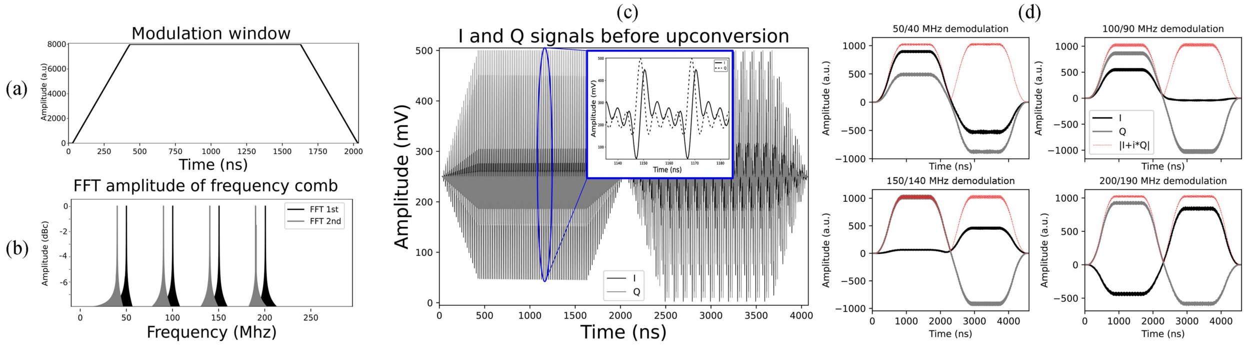


FIGURE 13. Hardware simulation of I/Q modulation and demodulation (Four frequencies 50 MHz spaced comb). (a) Modulation window: a trapezoidal pulse. (b) First (black) and second (gray) comb multiplexed. (c) Temporal digital output entering the DACs. The frequencies of the comb are shifted 10 MHz left after the first pulse is played. (d) Demodulated signal for the four frequencies.

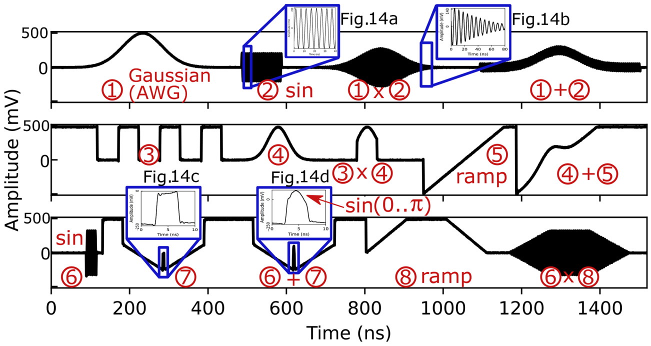


FIGURE 14. Measurements of three DAC channels with all mixing combinations ($N_{\text{out}} = 16$, 5 GS/s).

At the same time, the sine generator produces a frequency comb shown in Fig. 13(b). This comb consists of four frequencies starting at 50 MHz and spaced at 50 MHz intervals. Quickly changing sine parameters, in particular their frequencies, is useful in several cases, such as calibrating reflectometry to find resonances or driving qubits whose frequency can drift [36]. As an example, the comb is shifted left by 10 MHz between the first and second pulses.

The outputs of sine and ramp generators are then modulated to form the temporal signal, as shown in Fig. 13(c). I/Q upconversion, digital or analog, is not shown here. An additional delay is added to the temporal signal before demodulation to simulate cable propagation and DAC/ADC conversion time.

After demodulation, we recover our four symbols [see Fig. 13(d)]. Only 1/8 of the amplitude is recovered, which is expected since we have four sines, and the final demodulation step divides our signal by two. We successfully demodulated the original signal preserving its integrity.

B. DISCUSSION ON FASQUIC STRENGTHS

1) DISCUSSION ON CALIBRATION TIME

The strength of the system is its fast waveform reconfigurability capabilities. Fig. 12 is a typical example of a calibration process where the amplitude of the exchange part is optimized for gate fidelity. As shown in Section I, exchange time and amplitude are keys parameters to optimize for the high-fidelity implementation of SWAP gates. One way of calibrating the gates is to map the space of parameters. One way of doing this is to send a control sequence and measure the state of the qubit to observe whether or not the SWAP occurred. Since the SWAP is reversible there is no need for resetting the qubit between the steps and the procedure can continue with another set of parameters. By accumulating statistics or using methods, such as gradient descent, an optimal point in the design space can be found. One calibration step of the exchange amplitude or time only takes 76.8 ns of waveform reconfiguration with FASQuiC, which is far below current measurement time in spin qubits. An AWG-based architecture would need to rewrite half the waveform from a software layer taking, at a minimum, several microseconds. Even when using software feedback, FASQuiC does not need the compression step of COMPAQT nor the high bandwidth of classical AWG-based architectures. Therefore, FASQuiC provides a significant improvement in calibration time by closing the feedback loop faster.

Fig. 13 is also an example of calibration process of a frequency comb for optimizing driving or reflectometry-based measurement. It takes 352 ns through the AXI network to reconfigure the whole frequency comb, which can be done while the first comb is being played. Then, 44.8 ns are required to restart the CSGs (limited by the sine generator) and output the updated signal. These examples show how our design enables fast calibration of reflectometry or driving pulse control.

2) DISCUSSION ON SCALABILITY

FASQuiC was tested on a ZCU111 card; however, we have verified (with place and route) that on the larger ZCU216 card, it could be used to drive its 16 DACs. This is beyond the current most advanced spin qubit architecture [38]. There is sufficient logic in a ZCU216 card to drive up to 24 DACs, if they were available. There are two additional DACs compared with [25], while using a smaller FPGA [39]. Since the architecture is distributed, unitary resources given in Fig. 11

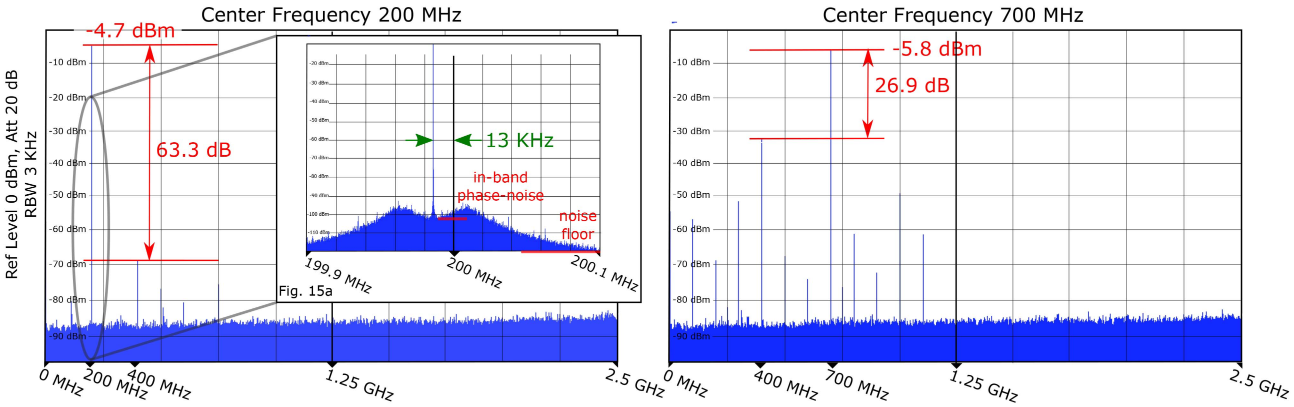


FIGURE 15. Measurement of 200 and 700 MHz frequencies generated by a sine generator with $N_{\mathrm{out}} = 8$. (a) Clear PN hump due to the local oscillator and in-band PN due to the phase detector of the PLL appear when zooming on the 200 MHz.

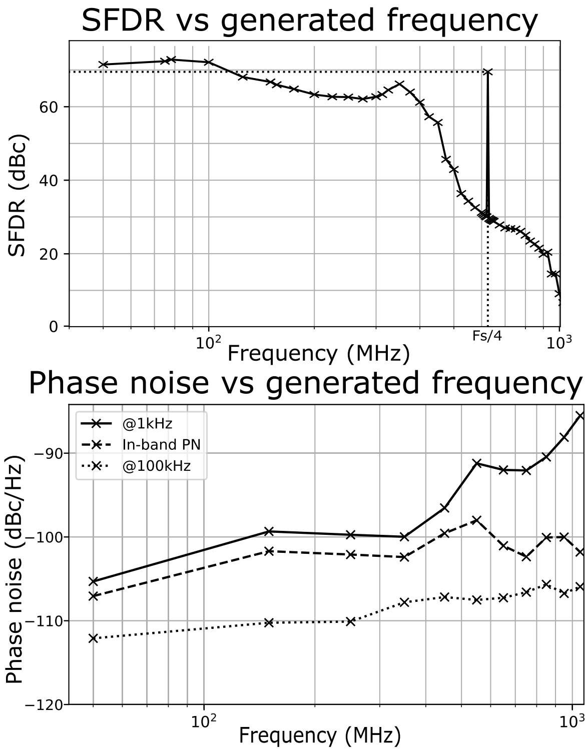


FIGURE 16. SFDR and PN measurement for three different offset frequencies at different generated frequencies from a sine generator at $N_{\mathrm{out}} = 8$.

can be safely multiplied by the number of DAC lanes, although larger configurations need slightly more logic for the AXI network. The feedback latency, given in Section III-A1, is independent of the number of channels.

In AWG architectures, the memory resources scale linearly with the DAC sampling rate, whereas in FASQuiC, the memory resources depend only on the complexity of the waveform.

With FASQuiC, when using driven control of spin qubits, digital multiplexing reduces the number of ac cables by up to 16×. Furthermore, with FASQuiC, combining analog and digital multiplexing, we can even drive up to 32 qubits through one ac cable using a single board, with space for six additional quasi-static controls.

On-the-fly generation reduces the memory cost for signal generation compared with an AWG-only based solution. In our design, AWG is only needed for tasks, such as pulse engineering of the exchange parts, lasting only a few nanoseconds. For instance, applying a SWAP gate to 24 pairs of spin qubits [38] would require 248 waveform parameters and instructions (1.2 kB) with FASQuiC, whereas AWG-based architectures would need 210 kB of data. Using COMPAQT DCT and RLE compression schemes for constant parts of the signal, while ensuring the same accuracy, this number drops to 55.9 kB which still represents 46.5× more data to transfer and store compared with FASQuiC.

C. EXPERIMENTAL RESULTS

1) EXPERIMENTAL SETUP

We use a Xilinx UltraScale+ RFSoC ZCU111, which provides versatile features for RF applications. It embeds 14-bit resolution RF-DACs (6.554 GS/s) and 12-bit resolution RF-ADC (4.096 GS/s). As mentioned earlier, the local oscillator has been changed and we added a 3-dB amplifier (LMH6554), which has a 2.8-GHz bandwidth at the output of the DACs. We also added relays at the output of the amplifier where measurements are performed (point γ in Fig. 1). Time-domain relative measurements were performed with a 23-GHz bandwidth and 100 GS/s scope. Frequency domain measurements were performed with a 13.6-GHz bandwidth RF-analyzer. We put an image rejection filter with 880-MHz cutoff frequency for measurements with $N_{\mathrm{out}} = 8$.

2) CSG VALIDATION

In this first experiment, we use seven DAC lines with an oversampling factor N_{out} of 16 to test all possible combinations of signal sources. Fig. 14 shows all mixing combinations for three DAC channels with double generators. The measurements are made at the output of the relays. Embedded synchronization of Xilinx DAC IP allows synchronization of DAC lanes at the output of the board.

The quality of our signals is mostly impacted by the output amplifier, which limits the bandwidth and could not be removed.

Power spectral density measurement at low frequency shows that RF-DACs from the ZCU111 have drifts around 9 $\mu\mathrm{V}/\sqrt{\mathrm{Hz}}$ at the second scale, which is a limiting factor for purely dc control of the system for long experiments running over several minutes. To use FASQUIC for dc control, alternative dc-DAC could be used and only fast control (quasi-static and RF-pulses) can be achieved with this implementation. We note that because of the fast waveform reconfiguration capabilities of FASQUIC, experiments run faster than on comparable solutions, relaxing the requirement for long-term dc stability.

3) FIGURES OF MERIT

Our two main figures of merit for the sine comb generation are the spurious free dynamic range (SFDR) and the PN.

PN is a random phase modulation around a perfect pure frequency, which translates to a random variation of the signal period called jitter. This frequency mismatch degrades the precision of rotations around the Bloch sphere when using pulse controls, which results in lower fidelity of quantum gates [40], [41]. For reflectometry, this mismatch reduces the signal-to-noise ratio when extracting the state of the qubits [42].

Deterministic modulation of the center frequency can also create spurious frequency tones, which reduces the fidelity of control by exciting unwanted resonances and reducing the coherency time of the qubits. The difference in power between the center frequency and the largest frequency spur corresponds to the SFDR, which should be maximized for the fidelity of the measurement and control of qubits.

4) SINE GENERATOR CHARACTERIZATION

In the second experiment, a CSG in the first configuration is initialized with $N_{\text{out}} = 8$, a ramp generator and a single sine generator. We can easily tune the amplitude of the comb by adjusting the height of the plateau output by the ramp generator [see Fig. 13(a)]. Specifically, when using I/Q modulation with on-board upconversion, the ZCU111 DAC IP requires half the points of the desired sampling rate, limiting us to $N_{\text{out}} = 8$.

Reflectometry typically works at frequencies of hundreds of MHz. A 200- and 700-MHz frequency analysis is shown in Fig. 15. The desired frequencies are shifted by 65 ppm (13 kHz shift for 200 MHz). This systematic shift is due to the 20 ppm LO that passes through the PLLs of the ZCU111, inducing a nonideal clock. A single correction factor can be applied to obtain more accurate frequency generation. The average frequency shift after correction is 1.8 kHz, which is in the range of the 4.7 kHz sine generator precision.

SFDR and PN measurements are shown in Fig. 16. The SFDR is above 60 dBc up to 350 MHz, which is far above the typical SFDR of upconversion frequencies [33]. The SFDR starts to drastically fall after 350 MHz and reaches 30 dBc at 600 MHz, a level that may no longer be acceptable. At 625 MHz, a quarter of the DAC sampling rate, there is a special point. At this point, the SFDR reaches 69 dBc because the output points are perfectly aligned with the contents of the lookup table for the single sine generator. So, the generator produces a high-quality sine wave.

The PN profile is shown in the 200-MHz zoom [see Fig. 15(a)]. A clear PN hump can be seen, this hump is due to the local oscillator noise and in-band PN of the phase detector of the PLL. This in-band PN is a major limiting factor for qubit control [41] and measurement [42], [43]. Our PN is measured at three offset frequencies: one before the band (at 1 kHz), another in the band, and the last after the band (at 100 kHz), close to the noise floor. PN slightly increases with the frequency generated but stays under −100 dBc/Hz for in-band PN up to 450 MHz.

Both measured SFDR and PN fulfill the requirements to control classical implementations of spin qubits [44].

In the third experiment, a CSG in the first configuration is initialized with $N_{\text{out}} = 8$, a ramp generator, and a 16-sine wave generator. The output frequency comb is shown in Fig. 17(a). The SFDR of the higher frequency sine (320 MHz) is reduced to 48 dBc compared with 65 dBc in the single sine generation case but its PN at 100 kHz does not increase. Indeed, with 16 sines, the power of each frequency is reduced and thus the parasitics at 500 and 750 MHz due to resonances of the amplifier are relatively larger.

5) MODULATION CHARACTERIZATION

Modulation from the DACs is now turned ON with CSGs in the third configuration at $N_{\text{out}} = 16$. An SSB upconversion to 1.25 GHz of the 16 frequency comb is presented in Fig. 17(b).

The upconversion slightly decreases the total power of the signal, by 3 dB. Even though more spurs appear in the spectrum, the SFDR is only degraded to 44.8 dBc (−3.2 dBc), and the PN at 100 kHz away from the 320 MHz peak is slightly increased to −107 dBc/Hz (+5 dBC/Hz).

6) SECOND NYQUIST ZONE CHARACTERIZATION

To test second Nyquist zone generation, CSGs in the third configuration are initialized at $N_{\text{out}} = 8$, to fit within the 2.8-GHz bandwidth of the output amplifier.

When we activate the second Nyquist zone, modulation can exceed half the sampling rate in the 1.25–2.5 GS/s zone. Fig. 17(c) shows an upconversion to 2 GHz of the 16 frequency combs. The activation of the second Nyquist zone increases the output power of the higher frequency [M1 in Fig. 17(c), 1.98 GHz] from −52.5 to

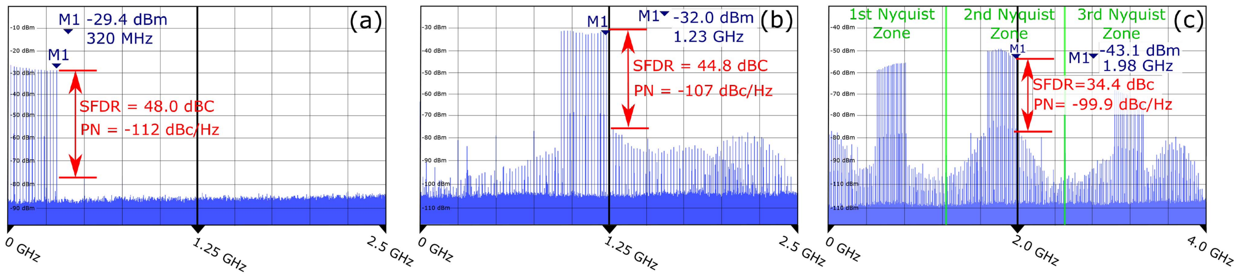


FIGURE 17. Spectral analysis of frequency comb generation with 3 kHz read bandwidth. (a) 16 frequency comb generated from a CSG in the first configuration with 20 MHz spacing and $N_{\text{out}} = 8$. (b) Modulation at 1.25 GHz of a 16 I/Q frequency comb with 20 MHz spacing and CSGs in third configuration at $N_{\text{out}} = 16$. (c) Modulation at 2 GHz of a 16 frequency comb with 20 MHz spacing and CSGs in third configuration at $N_{\text{out}} = 8$ using the second Nyquist zone technique.

−43.11 dBm. The second Nyquist zone technique is attractive for achieving higher frequency modulation but reduces comb generation quality: the SFDR in the second Nyquist zone is reduced to 34.4 dBc, and PN at a 100-kHz offset from the higher frequency peak is increased to −99.9 dBc/Hz.

IV. CONCLUSION

Our architecture achieves state-of-the-art qubit control, particularly optimized for the requirements of spin qubits by providing hardware for on-the-fly signal generation, which removes the need for huge memories to store precalculated data for AWGs. We demonstrate that on-the-fly signal generation can require 175× less memory than an approach based on AWGs. The low memory requirements and low feedback latency of our device would translate to a significant reduction in calibration time, which is a major overhead in LSQ computers.

Currently, a limiting factor for the scalability of LSQ computers is the number of cables in the refrigerator, which can be reduced with the SSB multiplexing of driving pulse control and readout provided by our design. FASQuiC can generate multiple ramps per cycle, providing subnanosecond control of quasi-static signals. This is essential for fine control of spin qubits, a requirement for high-fidelity quantum gates.

The requirements for quantum computers are continuously changing and one of the strengths of this design is its flexibility. Due to the parameterization of the RTL code, we can make use of bitstream switching. This reduces calibration time and allows FASQuiC to be reconfigured for the needs of each experiment.

The whole design has been tested and qualified on a Xilinx ZCU111 board showing high-quality generation of complex signals. Future versions of the FASQuiC controller will include arithmetic operations and conditional branching to enable more complex quantum feedback.

FASQuiC makes contributions to scalability through reduced memory usage and multiplexing, as well as providing state-of-the-art signal generation control all packaged in a flexible architecture, which constitute an important step toward spin qubit LSQ computers.

ACKNOWLEDGMENT

The authors thank Yvain Thonnart and Matias Urdampilleta for their theoretical guidance and Sylvain Dumas and Julien Minet for their help in the experimental setup.

REFERENCES

[1] J. Zhang, Y.-X. Liu, R.-B. Wu, K. Jacobs, and F. Nori, “Quantum feedback: Theory, experiments, and applications,” *Phys. Rep.*, vol. 679, pp. 1–60, Mar. 2017, doi: 10.1016/j.physrep.2017.02.003.

[2] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Phys. Rev. A*, vol. 86, Sep. 2012, Art. no. 032324, doi: 10.1103/PhysRevA.86.032324.

[3] D. Gottesman, “An introduction to quantum error correction and fault-tolerant quantum computation,” Apr. 2009, *arXiv:0904.2557*, doi: 10.48550/arXiv.0904.2557.

[4] N. Liyanage, Y. Wu, A. Deters, and L. Zhong, “Scalable quantum error correction for surface codes using FPGA,” in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Bellevue, WA, USA, 2023, pp. 916–927, doi: 10.1109/QCE57702.2023.00106.

[5] P. Stano and D. Loss, “Review of performance metrics of spin qubits in gated semiconducting nanostructures,” *Nature Rev. Phys.*, vol. 4, pp. 672–688, Oct. 2022, doi: 10.1038/s42254-022-00484-w.

[6] J. Yoneda et al., “A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%,” *Nature Nanotechnol.*, vol. 13, pp. 102–106, Feb. 2018, doi: 10.1038/s41565-017-0014-x.

[7] F. Arute et al., “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, pp. 505–510, Oct. 2019, doi: 10.1038/s41586-019-1666-5.

[8] M. Prathapan et al., “A system design approach toward integrated cryogenic quantum control systems,” in *Proc. IEEE 15th Workshop Low Temp. Electron.*, 2022, pp. 1–4, doi: 10.48550/arXiv.2211.02081.

[9] S. G. J. Philips et al., “Universal control of a six-qubit quantum processor in silicon,” *Nature*, vol. 609, pp. 919–924, 2022, doi: 10.1038/s41586-022-05117-x.

[10] M. F. Gonzalez-Zalba, S. de Franceschi, E. Charbon, T. Meunier, M. Vinet, and A. S. Dzurak, “Scaling silicon-based quantum computing using CMOS technology,” *Nature Electron.*, vol. 4, pp. 872–884, Dec. 2021, doi: 10.1038/s41928-021-00681-y.

[11] J. R. Petta et al., “Coherent manipulation of coupled electron spins in semiconductor quantum dots,” *Science*, vol. 309, pp. 2180–2184, Sep. 2005, doi: 10.1126/science.1116955.

[12] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen, “Spins in few-electron quantum dots,” *Rev. Modern Phys.*, vol. 79, pp. 1217–1265, Oct. 2007, doi: 10.1103/RevModPhys.79.1217.

[13] M. D. Shulman, O. E. Dial, S. P. Harvey, H. Bluhm, V. Umansky, and A. Yacoby, “Demonstration of entanglement of electrostatically coupled singlet-triplet qubits,” *Science*, vol. 336, pp. 202–205, Apr. 2012, doi: 10.1126/science.1217692.

[14] T. Meunier, V. E. Calado, and L. M. K. Vandersypen, “Efficient C-phase gate for single-spin qubits in quantum dots,” *Phys. Rev. B*, vol. 83, Mar. 2011, Art. no. 121403, doi: 10.1103/PhysRevB.83.121403.

[15] G. Burkard, T. D. Ladd, J. M. Nichol, A. Pan, and J. R. Petta, “Semiconductor spin qubits,” Dec. 2021, *arXiv:2112.08863*, doi: 10.48550/arXiv.2112.08863.

[16] P. Ao and J. Rammer, “Quantum dynamics of a two-state system in a dissipative environment,” *Phys. Rev. B*, vol. 43, pp. 5397–5418, Mar. 1991, doi: 10.1103/PhysRevB.43.5397.

[17] C. H. Yang et al., “Silicon qubit fidelities approaching incoherent noise limits via pulse engineering,” *Nature Electron.*, vol. 2, pp. 151–158, Apr. 2019, doi: 10.1038/s41928-019-0234-1.

[18] K. H. Park et al., “ICARUS-Q: A scalable RFSoC-based control system for superconducting quantum computers,” Dec. 2021, *arXiv:2112.02933*, doi: 10.48550/arXiv.2112.02933.

[19] Y. Yang et al., “FPGA-based electronic system for the control and readout of superconducting quantum processors,” *Rev. Sci. Instrum*, vol. 93, no. 7, Jul. 2022, Art. no. 074701, doi: 10.1063/5.0085467.

[20] Y. Xu et al., “QubiC: An open source FPGA-based control and measurement system for superconducting quantum information processors,” *IEEE Trans. Quantum Eng.*, vol. 2, 2021, Art. no. 6003811, doi: 10.1109/TQE.2021.3116540.

[21] S. Maurya and S. Tannu, “COMPAQT: Compressed waveform memory architecture for scalable qubit control,” in *Proc. 55th IEEE/ACM Int. Symp. Microarchitecture*, 2022, pp. 1059–1077, doi: 10.1109/MICRO56248.2022.00076.

[22] L. Stefanazzi et al., “The QICK (quantum instrumentation control kit): Readout and control for qubits and detectors,” Oct. 2021, *arXiv:2110.00557*, doi: 10.48550/arXiv.2110.00557.

[23] N. Irtija, J. Plusquellic, E. E. Tsiropoulou, J. Goldberg, D. Lobser, and D. Stick, “Design and analysis of digital communication within an SoC-based control system for trapped-ion quantum computing,” *IEEE Trans. Quantum Eng.*, vol. 4, 2023, Art no. 5500124, doi: 10.1109/TQE.2023.3238670.

[24] M. O. Tholén et al., “Measurement and control of a superconducting quantum processor with a fully-integrated radio-frequency system on a chip,” *Rev. Sci. Instruments*, vol. 93, Oct. 2022, Art. no. 104711, doi: 10.1063/5.0101398.

[25] N. Khammassi et al., “A scalable microarchitecture for efficient instruction-driven signal synthesis and coherent qubit control,” 2022, *arXiv:2205.06851*, doi: 10.48550/arXiv.2205.06851.

[26] M. Toubeix, E. Guthmuller, A. Evans, and T. Meunier, “A scalable low-latency FPGA architecture for spin qubit control through direct digital synthesis,” in *Proc. Des. Autom. Test Europe*, 2024. [Online]. Available: https://cea.hal.science/cea-04600892/

[27] J. K. Kim, J. Hwan Oh, J. H. Yang, and S. Eun Lee, “2D line draw hardware accelerator for tiny embedded processor in consumer electronics,” in *Proc. IEEE Int. Conf. Consum. Electron.*, 2019, pp. 1–2, doi: 10.1109/ICCE.2019.8661832.

[28] DDS compiler v6.0 product guide (pg141). Accessed: Jun. 18, 2024. [Online]. Available: https://docs.amd.com/

[29] F. Vigneau et al., “Probing quantum devices with radio-frequency reflectometry,” *Appl. Phys. Rev.*, vol. 10, no. 2, Jun. 2023, Art. no. 021305, doi: 10.1063/5.0088229.

[30] M. G. House et al., “Radio frequency measurements of tunnel couplings and singlet–triplet spin states in Si:P quantum dots,” *Nature Commun.*, vol. 6, Nov. 2015, Art. no. 8848, doi: 10.1038/ncomms9848.

[31] M. Urdampilleta et al., “Gate-based high fidelity spin readout in a CMOS device,” *Nature Nanotechnol.*, vol. 14, pp. 737–741, Aug. 2019, doi: 10.1038/s41565-019-0443-9.

[32] S. Schaal et al., “Fast gate-based readout of silicon quantum dots using Josephson parametric amplification,” *Phys. Rev. Lett.*, vol. 124, Feb. 2020, Art. no. 067701, doi: 10.1103/PhysRevLett.124.067701.

[33] U. Singhal et al., “SQ-CARS: A scalable quantum control and readout system,” *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 2006715, doi: 10.1109/TIM.2023.3305656.

[34] API reference – xlrd 2.0.1 documentation. Accessed: Jun. 18, 2024. [Online]. Available: https://xilinx.github.io/embeddedsw.github.io/rfdc/doc/html/api/index.html

[35] Zynq UltraScale RFSoC RF data converter v2.6 gen 1/2/3/DFE LogiCORE IP product guide (PG269). Accessed: Jun. 18, 2024. [Online]. Available: https://docs.amd.com/

[36] T. Nakajima et al., “Coherence of a driven electron spin qubit actively decoupled from quasistatic noise,” *Phys. Rev. X*, vol. 10, Mar. 2020, Art. no. 011060, doi: 10.1103/PhysRevX.10.011060.

[37] Y. Salathé et al., “Low-latency digital signal processing for feedback and feedforward in quantum computing and communication,” *Phys. Rev. Appl.*, vol. 9, Mar. 2018, Art. no. 034011, doi: 10.1103/PhysRevApplied.9.034011.

[38] F. Borsoi et al., “Shared control of a 16, semiconductor quantum dot crossbar array,” *Nature Nanotech.*, vol. 19, pp. 21–27, 2024, doi: 10.1038/s41565-023-01491-3.

[39] *Intel Stratix series FPGAs and SOCs*. Accessed: Jun. 18, 2024. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/stratix/10.html

[40] H. Ball, W. D. Oliver, and M. J. Biercuk, “The role of master clock stability in quantum information processing,” *npj Quantum Inf.*, vol. 2, Nov. 2016, Art. no. 16033, doi: 10.1038/npjqi.2016.33.

[41] J. Gong, Y. Chen, E. Charbon, F. Sebastiano, and M. Babaie, “A Cryo-CMOS oscillator with an automatic common-mode resonance calibration for quantum computing applications,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 69, no. 12, pp. 4810–4822, Dec. 2022, doi: 10.1109/TCSI.2022.3199997.

[42] Y. Peng, A. Ruffino, T.-Y. Yang, J. Michniewicz, M. F. Gonzalez-Zalba, and E. Charbon, “A Cryo-CMOS wideband quadrature receiver with frequency synthesizer for scalable multiplexed readout of silicon spin qubits,” *IEEE J. Solid-State Circuits*, vol. 57, no. 8, pp. 2374–2389, Aug. 2022, doi: 10.1109/JSSC.2022.3174605.

[43] B. Patra et al., “Cryo-CMOS circuits and systems for quantum computing applications,” *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 309–321, Jan. 2018, doi: 10.1109/JSSC.2017.2737549.

[44] J. P. G. van Dijk, B. Patra, S. Pellerano, E. Charbon, F. Sebastiano, and M. Babaie, “Designing a DDS-Based SoC for high-fidelity multi-qubit control,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 67, no. 12, pp. 5380–5393, Dec. 2020, doi: 10.1109/TCSI.2020.3019413.

Mathieu Toubeix (Graduate Student Member, IEEE) was born in France, in 1996. He received the Engineer’s degree specializing in electrical engineering and physics from Ecole Polytechnique, Palaiseau, France, in 2019, and the M.S. degree in electrical engineering specializing in microelectronics from Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2021. He is currently working toward the Ph.D. degree in physics with CEA, Grenoble, France, and Institut Néel, CNRS, Grenoble. His subject focuses on quantum control and the implementation of hardware feedback in the context of quantum computing with cryogenic spin qubits.

In 2021, he was with Thalès, TRT, Palaiseau for his master thesis on RISC-V multicore benchmarking. His research interests include the development of hardware at the interface of physics, especially quantum physics, hardware for high-performance computing, and ASIC implementation.

Eric Guthmuller received the master’s degree in engineering from Ecole Polytechnique, Palaiseau, France in 2009, the M.S. degree from Telecom Paris, Paris, France, in 2009, and the Ph.D. degree from University Pierre & Marie Curie, Paris, in 2013, both in computer science.

He joined the CEA-Leti, Grenoble, France, as a full-time Researcher until 2019, then within CEA-List. His main research interests include processor architectures and their memory hierarchy, and quantum computer control architectures.

Adrian Evans received the bachelor's degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1994, the DEA degree in microelectronics from the University of Paris VI, Paris, France, in 1995, and the Ph.D. degree in microelectronics reliability from the University of Grenoble, Grenoble, France, in 2014.

He has worked for 12 years on the design of complex digital ASICs for network processors with Cisco Systems and five years on circuits for space applications with IROC Technologies. Since 2017, he has been a Research Engineer with the CEA/List, Grenoble, where his research interests include efficient compute solutions for both embedded and high-performance applications.

Antoine Faurie received the B.S. degree in computer sciences from Université Jean Monnet, Saint-Etienne, France, in 2020, the engineering diploma degree in electronics from Télécom Saint-Etienne, Saint-Etienne, in 2023, and the M.S. degree in electronics for embedded systems from Université Claude Bernard, Lyon, France, in 2023. He is currently working toward the Ph.D. degree in qubit-array control electronics operating at cryogenic temperature with CEA, Grenoble, France.

From March to September 2023, he worked as a second-year M.S intern on an embedded software for measurement and control of quantum circuits with CEA. His research interests include cryo-electronics, radio-frequency (RF) engineering, and quantum computing.

Tristan Meunier received the bachelor's degree in fundamental mathematic from the University of Paris VI, Paris, France, in 1999, the M.S degree in fundamental physics from University Denis Diderot, Paris, France, in 2000, the DEA of Quantum Physics from Ecole doctorale de Physique de la région parisienne, in 2001 and the Ph.D. degree in induced quantum rabi oscillations from the University of Paris VI, France, in 2004.

He is a Senior CNRS Researcher with Institut Néel, Grenoble, France, and CTO with Quobly, Grenoble. Since 2005, he has participated to the world-wide effort on the coherent control of individual electron spins in semiconductors. Until the end of 2022, he was leading the quantum spin qubit Grenoble effort with CNRS. In December 2022, he co-founded the start-up Quobly, a spin-off of the CNRS-CEA effort on spin qubits in FDSOI technology. His research interests include the field of coherent control of individual quantum objects, both in atomic physics and solid state systems.

Dr. Meunier was the recipient of the Starting and Synergy Grants of the European Research Council on the coherent control of individual electron spins in semiconductor nanostructures, respectively, in 2012 and 2018 (synergy together with Maud Vinet and Silvano DeFrancheschi).