A Co-design Methodology for Scalable Quantum Processors and their Classical Electronic Interface

Jeroen van Dijk*†, Andrei Vladimirescu*‡§, Masoud Babaie*, Edoardo Charbon*†¶‖, Fabio Sebastiano*
* Delft University of Technology, Delft, The Netherlands, † Kavli Institute of Nanoscience, Delft,
The Netherlands, ‡ University of California, Berkeley, CA, § Institut Supérieur d'Electronique de Paris, Paris, France,
¶ Intel Corporation, Hillsboro, OR, ‖ EPFL, Lausanne, Switzerland

***Abstract*—A quantum computer fundamentally comprises a quantum processor and a classical controller. The classical electronic controller is used to correct and manipulate the qubits, the core components of a quantum processor. To enable quantum computers scalable to millions of qubits, as required in practical applications, the simultaneous optimization of both the classical electronic and quantum systems is needed. In this paper, a co-design methodology is proposed for obtaining an optimized qubit performance while considering practical trade-offs in the control circuits, such as power consumption, complexity, and cost. The SPINE (SPIN Emulator) toolset is introduced for the co-design and co-optimization of electronic/quantum systems. It comprises a circuit simulator enhanced with a Verilog-A model emulating the quantum behavior of single-electron spin qubits. Design examples show the effectiveness of the proposed methodology in the optimization, design and verification of a whole electronic/quantum system.**

I. Introduction

Quantum computing can potentially address problems that cannot be solved on classical computers within a reasonable time [1]. The generic architecture of a large-scale quantum computer is shown in Fig. 1 [2]. The actual quantum processor, at the bottom of the stack, must be cooled to cryogenic temperatures to exhibit quantum behavior. It is connected to the quantum-to-classical interface consisting of electrical circuits for the generation and read-out of signals to and from the quantum bits (qubits). The remaining upper layers, from the micro-architecture up to the algorithm layer, ensure proper algorithm execution by controlling the electronic hardware.

As state-of-the-art quantum processors contain less than 20 qubits [3], the classical electronic interface can be implemented with bulky, power-inefficient and expensive general-purpose room-temperature instruments connected via several wires to the qubits in the cryogenic chamber [4]. However, any useful quantum algorithm would require thousands or even millions of qubits [5], thus making the current approach unfeasible, because of cost, size, and wiring complexity.

As an alternative, cost-optimized and power-efficient electronic components can be designed and tailored to quantum processors. For scaling, it is essential that the control electronics are physically close to the quantum processor, possibly at cryogenic temperatures. However, the power budget of cryogenic electronics is restricted by the limited cooling power of cryogenic refrigerators [6], [7]. Therefore, it is necessary to assess the effect of any non-ideality of the electronic interface on the performance of the whole quantum computer, so as to consider practical trade-offs in classical circuits. In addition, such large-scale systems, spanning several technology domains, require appropriate verification methods and tools to enable a reliable design flow. Thus, a methodology and related tools for co-design of classical electronic and quantum systems are of paramount importance [8], and their introduction is urged by the fast pace at which larger quantum processors are currently developed [3].

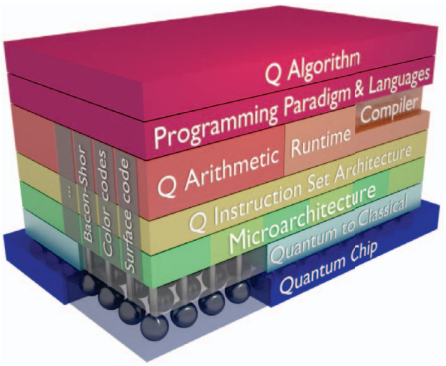


Fig. 1. The layered architecture of a quantum computer (taken from [2]).

Prior works focused on the upper layers in Fig. 1, down to the micro-architecture [2], [9]. For the layers below, electrical circuit simulators, such as SPICE and Spectre, are well-accepted industry standards and, equivalently, for the simulation of quantum systems, Hamiltonian solvers, such as QuTiP [10], are available. However, the actual interface between classical electronics and the quantum processor has mostly remained unexplored and, to the best of the authors' knowledge, no tool is available supporting both the simulation of classical electronics and quantum systems.

In this paper, a co-design methodology is proposed, along with a toolset called SPINE (SPIN Emulator) for the co-simulation of the electrical circuit and a quantum processor based on spin qubits. Using this tool, circuit designs can be optimized while ensuring qubit performance, and a verification of the entire quantum computer can be performed.

The paper is organized as follows: Section II introduces the proposed co-design methodology; Section III discusses the implementation of the proposed toolset; design examples are given in Section IV; possible future developments are discussed in Section V and conclusions are presented in Section VI.

978-3-9819263-0-9/DATE18/©2018 EDAA

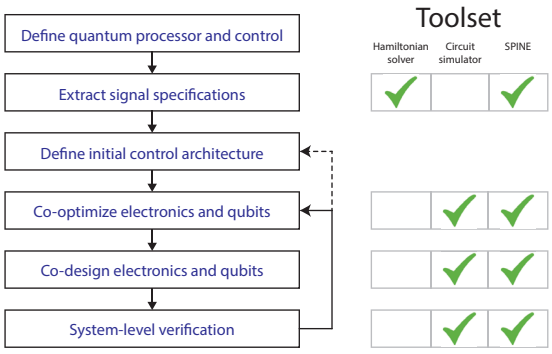


Fig. 2. Outline of the proposed classical electronic/quantum system co-design methodology along with the tools used in every step.

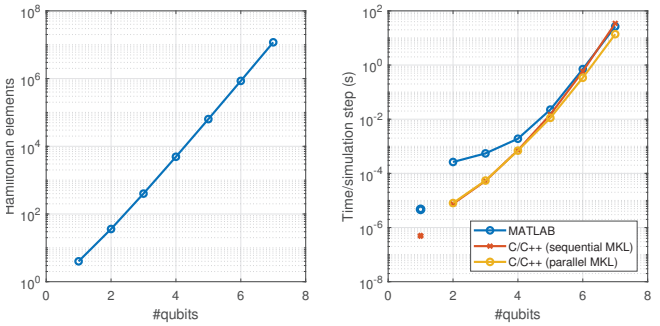


Fig. 3. The performance of the Hamiltonian simulators in MATLAB and C++. For C++ both implementations with and without multi-threading have been tested.

II. Co-design Methodology

The proposed methodology for the co-design of the classical electronic and quantum systems is summarized in Fig. 2. Every step of this process is aided by the introduced toolset, as exemplified in Section IV.

The design starts from an underlying qubit technology and control methods. Next, specifications for the control and read-out signals need to be extracted using qubit simulations such that the desired performance of the quantum processor is obtained (Section IV-A). Based on these results, trade-offs between qubit performance and performance of the control electronics, e.g. power and area, can be identified. With this information, an initial architecture can be defined and an error budget for the different circuit blocks can be drafted. For the chosen control architecture, further co-optimization of the electronics and the qubits can be performed, e.g., optimizing the number of qubits that can be frequency multiplexed over a single electronic channel [11]. Finally, the classical electronics and the quantum processor can be fully designed, and a system-level verification of the final design can be performed (Section IV-B).

III. Toolset Implementation

The SPINE toolset focuses on qubits in solid-state quantum dots, where the quantum information is encoded in the spin state of a single electron [4]. However, the toolset can be directly extended to other qubit technologies.

A. Hamiltonian Simulations

Following the proposed co-design methodology, simulations are required at various steps that do not necessarily entail the co-simulation of the electronics and the quantum physics. For such simulations, a generic Hamiltonian simulator can be used, which is also integrated with SPINE (Section III-B).

Hamiltonian simulations have been implemented both in MATLAB and C++. The optimized C++ implementation (with and without multi-threading) uses the Intel® Math Kernel Library (MKL). To benchmark these Hamiltonian simulators, an N-qubit system including singlet states (N electrons in N quantum dots) has been simulated, with a finite tunnel coupling between each pair of quantum dots. The results obtained on an Intel® Core™ i7-4700HQ with 8.0 GB DDR3 RAM are shown in Fig. 3[1]. It can be seen that the computation time scales with the size of the Hamiltonian under simulation. For small qubit systems, the C++ implementation is up to ∼35 times faster, and for larger systems almost twice as fast[2]. On our system, a 7-qubit Hamiltonian is the limit, showing a peak memory usage of 2.5 GB in MATLAB and 1.0 GB for the C++ implementation. However, as every qubit operation requires thousands of simulation steps, even fewer qubits can be simulated in practice to avoid excessive simulation times.

B. SPINE

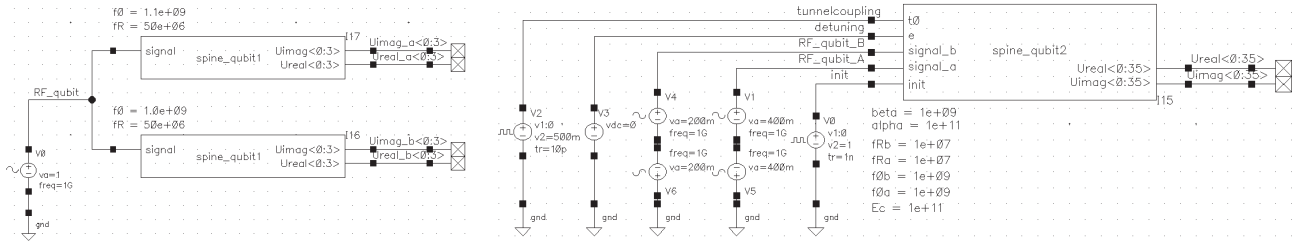
For the co-simulation of the classical electronics and quantum physics, SPINE was developed. As advanced electrical circuit simulators use quasi-static time-domain solvers, they provide a favorable environment for the inclusion of a time-discrete Hamiltonian simulation. Using Cadence® as a framework, the quantum physical system is included in the electrical simulation as a module that takes as input the control signals for the quantum system and outputs the quantum operation (see Fig. 4).

Since many circuit simulators support Verilog-A, this language is employed for the implementation of our module. The quantum state is updated during every time step of the electrical simulation. Different modules have been written for the different Hamiltonians, as required for a different number of qubits or energy levels. In SPINE, only modules emulating either one single-electron spin qubit (Fig. 4a) or a system of two coupled single-electron spin qubits (Fig. 4b) are currently available, but this can be easily expanded in the future.

The inputs to the Verilog-A blocks are the electrical signals applied to the quantum processor (see Fig. 4; `signal` is the RF-signal, `e` for detuning the quantum dots and `t0` to control the tunnel coupling) and `init` to reset the operation to the identity matrix. The resulting quantum operation U is available at the output with separated real (`Ureal<>`) and imaginary (`Uimag<>`) parts in row-major order. Parameters of the physical system are set as module parameters when

[1]The results shown for a single qubit system use an analytical solution for the matrix exponentiation step.

[2]MATLAB uses multi-threading.



(a) (b)

Fig. 4. The quantum physical system is included as a verilog-A module in the electrical circuit simulator. (a) Two modules, each emulating one single-electron spin qubit, have been instantiated; the two qubits are uncoupled and cannot be entangled. (b) A system of two coupled single-electron spin qubits.

instantiating the module in the circuit schematic (see Fig. 4, in which `Ec` is the charging energy of the quantum dot, `f0` the spin precession frequency, `fR` the rotation frequency at 1-V RF-signal, `beta` the tunnel coupling at 1 V and `alpha` the detuning energy at 1 V).

Due to the inclusion of the Hamiltonian simulation in the circuit simulation, time step control of transient simulations is managed by the circuit simulator, which relaxes the time step when tolerable. This speeds up both the circuit simulation and the Hamiltonian simulation while also recording and taking into account in the quantum simulation events occurring on a very short timescale, such as glitches. A maximum time step is set by the Verilog-A module to ensure accurate simulation of the quantum physics.

IV. Design Examples

A. Optimization of Power Consumption

A major concern in scaling quantum computers is the power consumption required by the control electronics. A reduction of the power consumption can be obtained at the cost of quality of the signal being generated for qubit control, or more errors during qubit read-out. Quantum simulations considering signal non-idealities are required in order to assess the minimum signal quality that can ensure a tolerable error rate in the quantum processor. Once the effect of errors on the qubit fidelity is known, a larger error budget can be allocated to the sources of error that require more power in the electrical circuit to be mitigated, thereby optimizing the total power consumption while ensuring the target qubit fidelity.

As an example, we will consider the microwave control signal required to perform a qubit rotation and focus on deriving the effect of inaccuracies in the amplitude. To determine the accuracy requirement, simulations of the quantum system have been performed for control signals with different amounts of amplitude errors on an otherwise ideal signal, as shown in Fig. 5. For specific cases, such as the amplitude accuracy shown here, these requirements can also be derived analytically [12], [13]. After a proper budgeting of the various errors, control circuits can be designed meeting the specifications and can be validated using a co-simulation of the electronics and physical system. As an example, Fig. 6a shows an output-driver circuit designed in a standard CMOS technology that

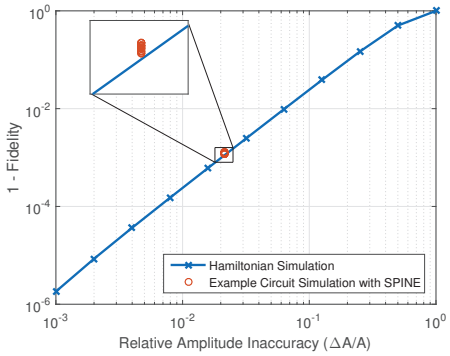
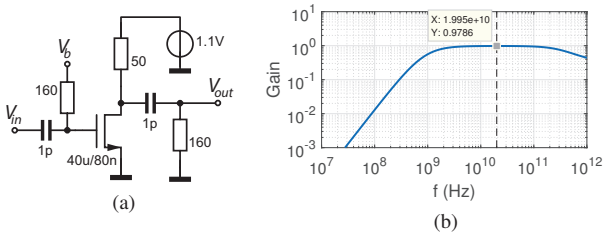


Fig. 5. The simulated fidelity for one of the sources of inaccuracy in the microwave amplitude.



(a) (b)

Fig. 6. (a) An example output driver circuit. (b) The simulated gain versus frequency.

can be used to apply the microwave signal to the quantum processor. This circuit has a gain lower than unity (Fig. 6b) resulting in an amplitude inaccuracy and a reduced control fidelity. The fidelity obtained using a co-simulation of this electrical circuit in Spectre with SPINE is plotted in Fig. 5, which is in good agreement with the results obtained from the Hamiltonian simulation. Multiple points are visible in the plot, reflecting the different runs of the transient simulation with noise.

B. System Level Verification

To show the power of the proposed toolset for the verification of the quantum computer, a full system comprising a high-level description of the quantum computer's controller, Verilog-A models of the digital-to-analog converters (DACs) and an analog mixer circuit has been integrated together with SPINE (Fig. 7). The performance was verified by simulating

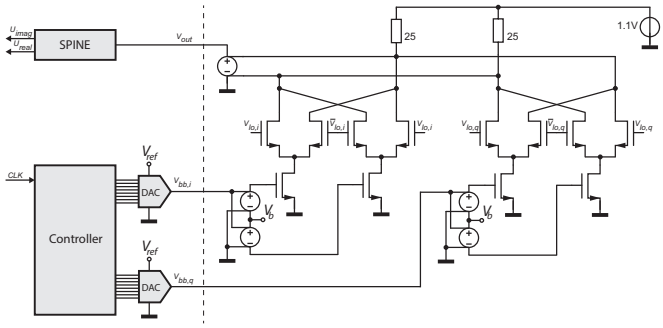


Fig. 7. Schematic of a full system, containing a high-level description of the controller, Verilog models for the DACs, an analog mixer circuit and finally SPINE.

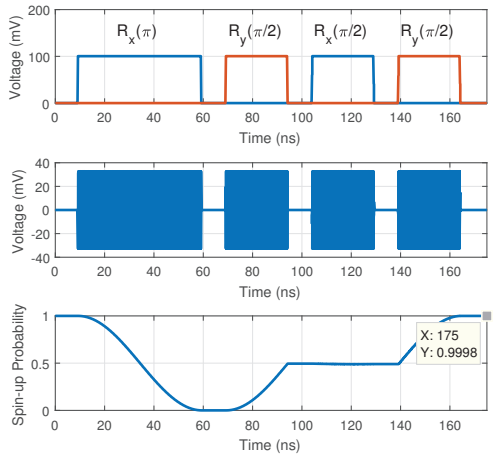


Fig. 8. Result of the full system simulation, from top to bottom: the voltage at the output of the DACs, the I- and Q-signals, driving the mixer, along with a description of the quantum gate; the voltage at the output of the mixer, driving the qubit; and finally the qubit spin-up probability assuming the qubit was initialized to spin-up.

a small quantum algorithm consisting of 4 gates, see Fig. 8. It can be seen that in response to the controller, the DACs generate the required in-phase and quadrature-phase signals for the mixer, and the analog circuit performs the required upconversion. In response to this signal, the qubit performs the expected rotations, finally achieving a 99.98% probability of success.

V. Perspective

A first step has been made towards the co-simulation of the classical electronics and quantum physics. The proposed toolset is used all throughout the proposed co-design methodology, and is in fact currently used in the design of a complex cryogenic integrated circuit for the control of spin qubits [13]. The toolset is used to evaluate the effect of signal non-idealities, both noise and inaccuracies, in the various control signals required for performing operations on single or multiple qubits.

However, there are still several opportunities for the design automation community to enhance this toolset. For instance, tremendous speed-ups are required to enable the simulation of larger qubit systems, e.g., to facilitate the inclusion of higher levels of the quantum computer (Fig. 1). Next, modules can be added or enhanced, both for the simulation of other quantum technologies and for simulating changes in electrical properties depending on qubit states, which will be the basis for simulating qubit read-out.

VI. Conclusion

A tailor-made electronic interface for quantum processors is required to enable the scaling of a quantum computer to the size required for any practical application. Co-design of the classical electronics and the quantum processor is essential to obtain a full system that meets the required performance under practical constraints, such as cost, size, power and reliability. In this paper, we have proposed a co-design methodology and the related toolset that meets such demand, and we have demonstrated, via practical examples, how it can be applied to co-optimize, co-design and verify the classical electronic/quantum systems. While we are verifying the factual effectiveness of the proposed approach through the design of a complex cryogenic integrated circuit for the control of spin qubits, this already represents the first fundamental step towards a holistic co-design platform.

Acknowledgment

The authors would like to thank Intel Corp. for funding.

References

[1] A. Montanaro, "Quantum algorithms: an overview," *npj Quantum Information*, vol. 2, pp. 15 023 EP –, Jan 2016, review Article.

[2] X. Fu *et al.*, "An experimental microarchitecture for a superconducting quantum processor," *arXiv preprint arXiv:1708.07677*, 2017.

[3] T. Watson, "IBM doubles compute power for quantum systems, developers execute 300k+ experiments on IBM Quantum Cloud," May 2017. [Online]. Available: https://developer.ibm.com/dwblog/2017/quantum-computing-16-qubit-processor

[4] M. Veldhorst *et al.*, "A two-qubit logic gate in silicon," *Nature*, vol. 526, no. 7573, pp. 410–414, 2015.

[5] A. G. Fowler *et al.*, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A*, vol. 86, p. 032324, Sep 2012.

[6] J. Hornibrook *et al.*, "Cryogenic control architecture for large-scale quantum computing," *Physical Review Applied*, vol. 3, no. 2, p. 024010, 2015.

[7] E. Charbon *et al.*, "Cryo-CMOS circuits and systems for scalable quantum computing," in *Solid-State Circuits Conference (ISSCC), 2017 IEEE International*. IEEE, 2017, pp. 264–265.

[8] F. Sebastiano *et al.*, "Cryo-CMOS electronic control for scalable quantum computing: Invited," in *Proceedings of the 54th Annual Design Automation Conference 2017*, ser. DAC '17. New York, NY, USA: ACM, 2017, pp. 13:1–13:6.

[9] N. Khammassi *et al.*, "QX: A high-performance quantum computer simulation platform," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 464–469.

[10] J. Johansson *et al.*, "QuTiP: An open-source python framework for the dynamics of open quantum systems," *Computer Physics Communications*, vol. 183, no. 8, pp. 1760–1772, 2012.

[11] M. Steffen *et al.*, "Simultaneous soft pulses applied at nearby frequencies," *Journal of Magnetic Resonance*, vol. 146, no. 2, pp. 369–374, oct 2000.

[12] H. Ball *et al.*, "The role of master clock stability in quantum information processing," *npj Quantum Information*, vol. 2, pp. 16 033 EP –, Nov 2016, review Article.

[13] J. van Dijk *et al.*, "Trade-offs in engineering a scalable cryogenic controller for solid-state spin-qubits," in *International Workshop on Silicon Quantum Electronics*, Aug. 2017.