Quantum Inspire: QuTech's platform for co-development and collaboration in quantum computing

Last, Thorsten, Samkharadze, Nodar, Eendebak, Pieter, Versluis, Richard, Xue, Xiao, et al.

Thorsten Last, Nodar Samkharadze, Pieter Eendebak, Richard Versluis, Xiao Xue, Amir Sammak, Delphine Brousse, Kelvin Loh, Henk Polinder, Giordano Scappucci, Menno Veldhorst, Lieven Vandersypen, Klará Maturová, Jeremy Veltin, Garrelt Alberts, "Quantum Inspire: QuTech's platform for co-development and collaboration in quantum computing," Proc. SPIE 11324, Novel Patterning Technologies for Semiconductors, MEMS/NEMS and MOEMS 2020, 113240J (23 March 2020); doi: 10.1117/12.2551853

SPIE.

Event: SPIE Advanced Lithography, 2020, San Jose, California, United States

Quantum Inspire – QuTech's platform for co-development and collaboration in Quantum Computing

Thorsten Last[1,2], Nodar Samkharadze[1,2], Pieter Eendebak[1,2], Richard Versluis[1,2], Xiao Xue[1,3], Amir Sammak[1,2], Delphine Brousse[1,2], Kelvin Loh[2], Henk Polinder[2], Giordano Scappucci[1,3], Menno Veldhorst[1,3], Lieven Vandersypen[1,3], Klára Maturová[1,2], Jeremy Veltin[2], Garrelt Alberts[1,2]

[1]QuTech Advanced Research Center, Lorentzweg 1, 2628 CJ Delft, The Netherlands
[2]TNO – Netherlands Organization for Applied Scientific Research, Stieltjesweg 1, 2628 CK Delft, Netherlands
[3]Kavli Institute of Nanoscience, TU Delft, Lorentzweg 1, 2628 CJ Delft, The Netherlands

ABSTRACT

The mission of QuTech is to bring quantum technology to industry and society by translating fundamental scientific research into applied research. To this end we are developing Quantum Inspire (QI), a full-stack quantum computer prototype for future co-development and collaborative R&D in quantum computing.

A prerelease of this prototype system is already offering the public cloud-based access to QuTech technologies such as a programmable quantum computer simulator (with up to 31 qubits) and tutorials and user background knowledge on quantum information science (www.quantum-inspire.com). Access to a programmable CMOS-compatible Silicon spin qubit-based quantum processor will be provided in the next deployment phase. The first generation of QI's quantum processors consists of a double quantum dot hosted in an in-house grown SiGe/^{28}Si/SiGe heterostructure, and defined with a single layer of Al gates.

Here we give an overview of important aspects of the QI full-stack. We illustrate QI's modular system architecture and we will touch on parts of the manufacturing and electrical characterization of its first generation two spin qubit quantum processor unit. We close with a section on QI's qubit calibration framework. The definition of a single qubit Pauli X gate is chosen as concrete example of the matching of an experiment to a component of the circuit model for quantum computation.

Keywords: Quantum Computer, Quantum Device Manufacturing, Spin Qubit in Silicon, CMOS

Novel Patterning Technologies for Semiconductors, MEMS/NEMS and MOEMS 2020,
edited by Martha I. Sanchez, Eric M. Panning, Proc. of SPIE Vol. 11324, 113240J
 · doi: 10.1117/12.2551853

* Primary contact: thorsten.last@tno.nl, phone +31 6 53249155

1. INTRODUCTION

A future computation ecosystem consisting of classical supercomputers and general-purpose quantum computers has the potential to tackle big problems facing society in energy, health and security [1, 2]. Until then traditional scaling can ensure continuous growth of processing power for the foreseeable future [3]. Inserting quantum computing (QC) however into such an ecosystem would mean a major shift from classical computation with its difference in system architecture, system requirements and different way of programming the quantum processing unit [4, 5].

QC's potency stems from its fundamental building block, the quantum bit (qubit) [6]. Unlike classical bits, qubits possess quantum mechanical resources such as superposition and entanglement for computation. Superposition is a characteristic of single qubits in which a qubit can exist in a linear combination of distinct quantum states. Entangled qubits form new inseparable multi-qubit states. Exploiting these resources for computation could give one the opportunity to solve problems which are out of reach for classical computers [1, 2, 7]. But note, the number of qubits in a quantum processing unit need to increase manifold before this quantum advantage becomes reality [1]. This large component-level as well as system-level challenge is accompanied by another even more crucial aspect of qubits, their susceptibility to errors. This susceptibility to errors is called decoherence. The decoherence process leads to errors in the computation. It is extremely difficult to prevent this from happening. The error probability per operation must be below a so-called fault-tolerance threshold for each and every qubit [8, 9]. Errors above this threshold are still commonly unavoidable in current quantum processors. But even for these systems with error-prone processors major efforts are currently undertaken to find useful applications [2]. The number of use cases in the realm of quantum simulations, optimization and machine learning is steadily growing.

At QuTech we are convinced that offering the public broad access to technologies such as quantum computing hardware as well as a quantum computing simulator will further stimulate this growth. For this reason we develop Quantum Inspire (QI), our prototype platform for quantum computation. Quantum Inspire comprises of a number of layers including quantum hardware, classical control electronics, and a software front-end with a cloud-accessible web-interface. Such a system is called a full-stack. Full-stack systems are essential test beds for understanding this novel computational paradigm. They can act as technology accelerators because only through careful analysis of the individual system layers and their interdependencies is it possible to detect the gaps and necessary next steps in the innovation roadmap and supply chain. QI, in general, is designed to be a platform with a high degree of modularity. And thus we hope QI can foster innovation in quantum computing through collaboration and co-development.

The following sections give insights into key aspects of QI's system architecture, the signal flow through the full-stack (Section 2), its CMOS-compatible processing unit (Section 3), and will conclude with an example of qubit calibration and how to define the Pauli X gate, an essential quantum gate (Section 4). We will close with a summary and an outlook (Section 5).

2. ASPECT OF QI'S ARCHITECTURE, ITS SOFTWARE FRONT-END & QX SIMULATOR

In its current form, programming a quantum computer is different from programming a classical computer. Therefore QI is already offering the public cloud-based access to QuTech's QX universal quantum computing simulator [10]. QI's user experience comes with a website www.quantum-inspire.com and a software development kit (SDK) on which the user has a variety of ways to program algorithms, execute these algorithms and examine the results.

Currently two QX simulator back-ends are in use, differing only in the usable memory and with that in the number of ideal qubits which can be simulated. The largest memory can be obtained by running the simulator on SURF's Dutch National Supercomputer Cartesius. There, currently, a quantum computer with up to 31 qubits can be emulated on a single node. The results of the algorithm are returned to the user via the web-app or SDK.

Quantum Inspire's programming language in which quantum algorithms are written and executed is called cQASM. cQASM was developed by the Quantum & Computer Engineering group at TU Delft and is a variant of the Quantum Assembly Language QASM [10]. Algorithms can be programmed using the QI Editor or using the SDK which provides a thin layer between the QI application programming interface (API) and other programming platforms using Python, namely ProjectQ [11] and QisKit [12].

In the future the system can be upgraded to include hardware back-ends such as a CMOS-compatible quantum processor. Important aspects of this processor will be covered in the following sections 3 and 4. Then, when executing a quantum algorithm in cQASM through the QI web-interface or API, the algorithm will be compiled into micro-instructions that will be generated and will steer the digital and analog control signals that are required to execute the algorithm. From there conventional DC-carrying cables and high frequency transmission lines will bring the required signals to and from the quantum processor which is placed on a printed circuit board (PCB) at the 10 mK stage of a dilution refrigerator.

A snapshot of the QI website is shown in Figure 1 (a). A schematic representation of the system architecture can be found in Figure 1 (b). And Figure 1 (c) shows parts of the hardware back-end in the Quantum Inspire lab at QuTech in Delft.

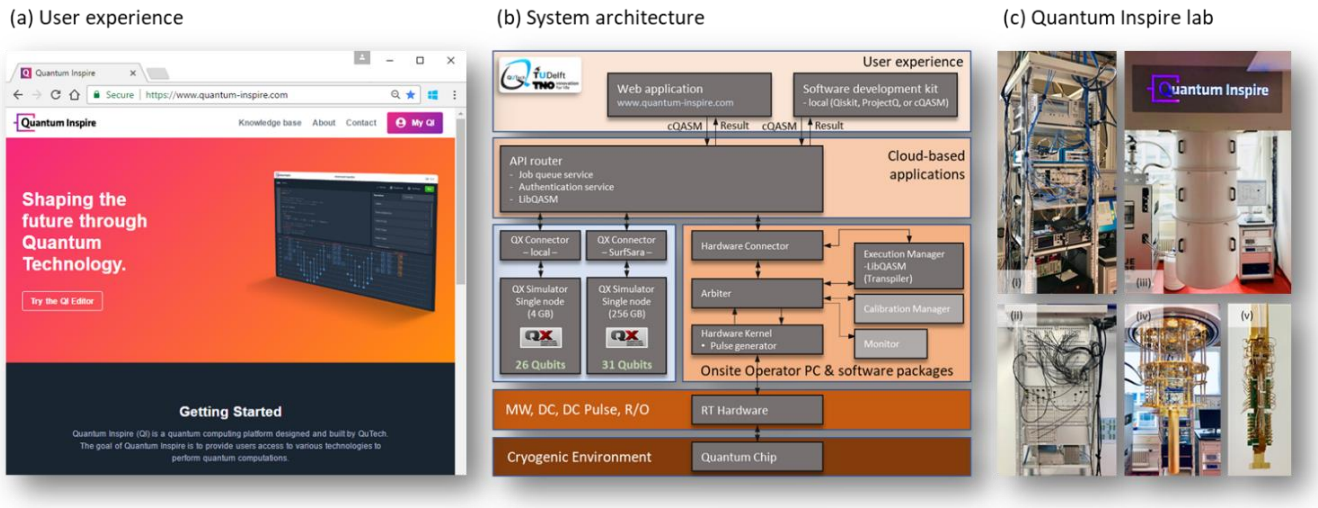

Figure 1: (a) Snapshot of QI's web interface at www.quantum-inspire.com. (b) Illustration of the main components of Quantum Inspire's full-stack quantum computer prototype system. A prerelease of this prototype system is already offering the public cloud-based access to QuTech's QX quantum computer simulator with up to 31 ideal qubits. In figure (c) the hardware back-end in the Quantum Inspire lab at QuTech, Delft, is presented. The control and measurement electronics (i), (ii) consists in part of off-the-shelf electronics and in part of in-house components. (iii) shows the fully assembled dilution refrigerator. The inside of the dilution refrigerator can be seen in (iv) and (v).

3. CMOS-COMPATIBLE SPIN QUBITS IN SI-BASED QUANTUM DOTS

Qubits had been prototyped and demonstrated on many different platforms including trapped ions, neutral atoms, solid-state devices, photons, and Nitrogen-Vacancy centers in Diamond [13]. In all of these implementations the basic sequences of quantum processing, namely initialization, single and two-qubit control and state read-out had been shown. Solid-state qubits which include superconducting Transmons and electron spins trapped in semiconductor quantum dots are particularly promising building blocks of a quantum processor [14, 15]. These devices can be manufactured and tailored by standard lithographic techniques, which is a considerable advantage for potential future large-scale integration of a large number of qubits. In addition, they are controlled and read out by current-state electronics. Superconducting Transmon technology is more mature at this point in time, with current state of the art devices incorporating up to 50 qubits, heralding the NISQ era (Noisy Intermediate-Scale Quantum era [2]). Spin qubits, on the other hand, are very promising from the point of view of very large-scale integration, thanks to their small size and similarity to classical transistor technology. Quantum Inspire's modularity ensures that different qubit implementations can be added as hardware back-ends.

In the following sections we will touch on our first generation two spin qubit quantum processor in a Silicon-based double quantum dot. Spin qubits in semiconductor quantum dots have been thoroughly studied for almost two decades [14, 16-19]. In principle they are formed as follows: An electron needs to be confined to a small region of space (few tens of nanometer in diameter) at a semiconductor/semiconductor or semiconductor/dielectric interface. Such a small region of space is called quantum dot, and results in a discrete spectrum of electronic orbital energies (resolved at milli-Kelvin temperatures). The confinement can be achieved by a combination of semiconductor band offsets and electrostatic metal

gates on top of the semiconductor (see Figure 2 (b), (c)). Then the electron ground state of the single electron confined in this quantum dot region is (Zeeman-) split into two levels by an externally applied magnetic field, hence, forming a canonical quantum two-level system (or qubit) with well-defined ground (spin-down) and excited (spin-up) states.

In subsection 3.1 we will describe the fabrication process. Subsection 3.2 will focus on results of the electrical qualification of our quantum devices. In subsection 3.3 we will shortly address our initial work on implementing a Schrödinger/Poisson solver which in the future could support designing next generations of devices.

3.1 Quantum Device Manufacturing

In general, a processing chip which is considered to be a component of a full-stack system architecture asks for more stringent specifications in electronic stability and robustness than required for proof-of-principle scientific experiments. Taking this functional requirement into account we developed a thorough device development feedback loop including design, materials, fabrication and electrical screening. Figure 2 (a) shows the typical lifecycle of a quantum device from crystal growth to the QI, split into 4 development modules.

In the first module, an undoped ^{28}Si quantum well heterostructure is grown with reduced-pressure chemical vapor deposition (RP-CVD). Starting with an n-type Si(100) wafer, first a 900 nm linearly graded Si_xGe_{1-x} layer up to 30% Ge content, followed by a 400 nm $Si_{0.7}Ge_{0.3}$ buffer is grown (Fig. 2 (b)-1) as a virtual substrate. The growth is then continued by a 10 nm strained ^{28}Si layer (Fig. 2 (b)-2), a 30 nm $Si_{0.7}Ge_{0.3}$ spacer (Fig. 2 (b)-3), and a 1 nm Si cap. The conduction band of Si lies 0.14 eV below that of $Si_{0.7}Ge_{0.3}$, which effectively confines the accumulated electrons in two dimensions inside the ^{28}Si layer, separated from disordered amorphous dielectrics used to electrically insulate the gates from the crystalline $Si_{0.7}Ge_{0.3}$ spacer. In the second module, a full 4" wafer is processed with optical lithography to create n++ doped areas providing Ohmic access to the quantum well layer (Fig. 2 (b)-4), and a 7 nm Al_2O_3 gate dielectric deposited with a thermal atomic layer deposition (ALD) process (Fig. 2 (b)-5). On top of the n++ implanted areas windows in the Al_2O_3 are opened with HF and 5/45 nm Ti/Pd contact pads are deposited (Fig. 2 (b)-6). Next the wafer is diced into 52 1x1 cm^2 dies, to be individually processed with electron beam lithography. In the third module, 1x1 cm^2 dies are processed with electron beam lithography and a lift-off process, to create 25 nm thick Al nanogates (Fig. 2 (c)) with fanout to the bond pads, and fanout of the Ohmic contacts to the bond pads (Fig. 2 (b)-7). Then a second layer of 10 nm ALD Al_2O_3 is deposited on the chip to insulate the nanogates (Fig. 2 (b)-8), which is followed by two more e-beam lift-off steps to create a 50 nm thick Al top gate (Fig. 2 (b)-9) and a 200 nm Co nanomagnet (Fig. 2 (b)-10) (dashed outline on Fig. 2 (c)). After nanofabrication, the dies are diced into 5x5 cm^2 individual devices, which are glued and wire-bonded to a PCB. This concludes the fabrication process.

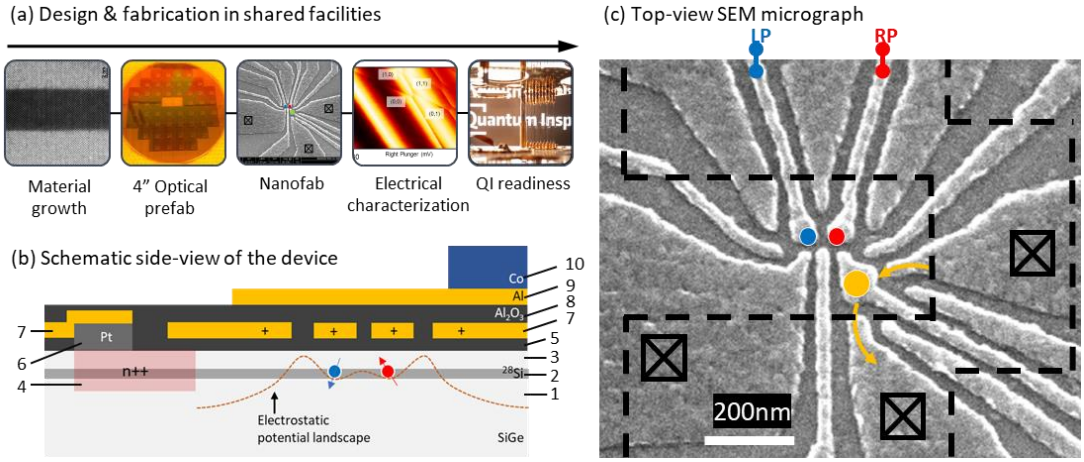

Figure 2: The design and fabrication of the ^{28}Si-based two spin qubit quantum processor: In (a) the device development feedback loop is presented including materials growth, pre- and nanofabrication of the device, with subsequent electrical screening. The optical CMOS-based prefab step for the chip periphery ensures an increased device turn-around and fast process development learning. The nanoscale part of the devices is fabricated with electron beam lithography. Only after thorough electrical screening QI readiness of the chips is determined. (b) A schematic side-view cross-cut of a typical quantum device design.

3.2 Electrical characterization: Charge stability and single-shot readout

In the fourth module electrical testing of the devices is conducted to determine their usability for quantum information processing. Here only key steps are illustrated. For detailed descriptions on quantum transport and spin qubits in quantum dot devices in general we would like to refer to the following comprehensive overview articles [14, 16-19].

The electrical characterization starts with the cool-down of the wire-bonded devices in a dilution refrigerator below a temperature of 20 milli-Kelvin to suppress thermal fluctuations. Then voltages are applied to the nanogates to form quantum dots and a single electron transistor (SET) in the ^{28}Si quantum well. First, positive voltages are applied to the three accumulation gates which will provide an electron reservoir for loading single electrons into the quantum dots, and source and drain for the SET. Subsequently voltages are applied to the other gates to form two quantum dots to be used as qubits (Fig. 2 (c), red and blue circles) and an SET (Fig. 2 (c), orange circle). In the quantum dots and the SET, motion of electrons is confined in all three spatial directions, and energy levels are quantized. In addition, in order to charge the quantum dot island with additional electrons, the charging energy must be overcome. Only when the electrochemical potential corresponding to a transition between N and N+1 electrons on the island is aligned within the window between the source and drain electrochemical potentials, a current can flow through the SET, following the path indicated with the orange arrows on Figure 2 (c). Since the electrochemical potential in the SET is affected by the electrostatic potential, the conductance of the SET can be used as a sensitive probe of the electrostatic landscape around the SET.

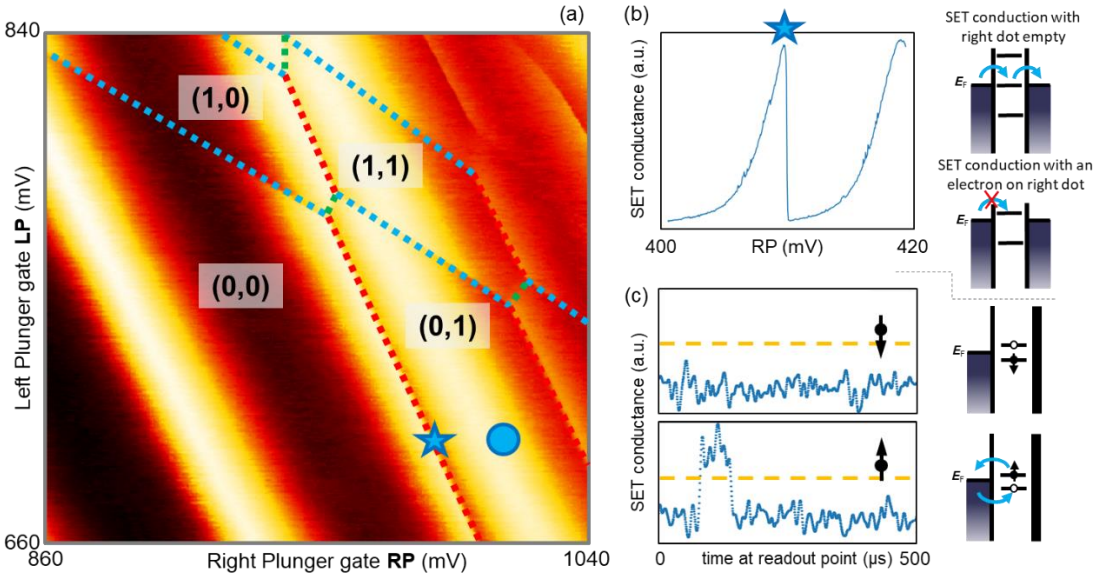

Figure 3: (a) Areas of different charge occupancies in the few electron regime of the charge stability diagram for a double dot. (b) Typical response of the SET conductance as a function of the plunger gate of the right dot. The SET is tuned to maximize the change in its conductance at the transition point. (c) Typical time dependent SET signal traces at the readout point, indicating spin-down or spin-up electron.

Figure 3 (a) shows a charge stability diagram, where the voltages on the left and right quantum dot plungers (Fig. 2 (c), gates marked with "**LP**" and "**RP**") are swept, while monitoring the SET conductance with a radio frequency tank circuit. A charge stability diagram can be viewed as the landscape of qubit operation. Bright yellow high conductance bands on Figure 3 (a) indicate direct cross-capacitance between **LP** and **RP** and the SET. Besides the direct influence of **LP** and **RP** on the SET resistance, abrupt jumps can be observed in the SET conductance, called charge transition lines, indicating a single electron being loaded into the left quantum dot (Fig. 3 (a), blue dotted lines), right quantum dot (Fig. 3 (a), red dotted lines), or an electron moving between the left and right quantum dots (Fig. 3 (a), green dotted lines). In the lower left quadrant of the charge stability diagram no more charge transition lines are observed, which means that the quantum dots are fully depleted. By ramping up **LP** and **RP** voltages and counting the number of left and right charge transition lines crossed, one can identify regions labelled **(0,0)**, **(0,1)**, **(1,0)**, **(1,1)**, where the digits denote the number of electrons trapped on the left and right dots respectively. Finally, the rest of the gates are used to control tunnel barriers between the dots, and between dots and reservoirs. The requirements for the electrical performance of the devices to be used in quantum

information processing: the construction of a charge stability diagram in a single-electron regime, similar to the one shown on Fig. 3 (a), with stable charge transition lines, and controllable tunnel couplings between the dots and the reservoirs.

In the following, we briefly describe operations of a quantum computer on an example of a single spin qubit hosted in the right quantum dot. The charge stability diagram illustrates the area in **LP/RP** gate space in which we can navigate to start calibrating a spin qubit. For this, we choose two configurations in gate space illustrated in Fig. 3 (a), for qubit manipulation (marked with a circle) and for qubit readout and initialization (marked with a star). **LP** and **RP** gates are equipped with high-bandwidth coaxial line connections, to which arbitrary waveforms can be applied, allowing fast navigation in the gate space with sub-nanosecond resolution. Each measurement cycle consists of initialization, manipulation and readout, where the gate configuration is pulsed from star to circle and back to star position on the charge stability diagram. The qubit manipulation point is chosen in the middle of the (0,1) region, where the electron is well isolated from the reservoir. To define the computational states of the qubit, an external magnetic field is applied to the device. For a single-electron spin qubit the computational states will be "spin-down" - ground state of the electron spin where it is aligned with the external magnetic field, and "spin-up" - excited state of the electron spin where it is anti-aligned with the external magnetic field. The energy difference between the spin-up and spin-down states is directly proportional to the external magnetic field, and is chosen to be $E/h \sim 15.66$ GHz. This is the frequency which we need to drive transitions between the two states of the qubit. More details on qubit manipulations can be found in section 4. The energy difference between spin-down and spin-up can also be used to measure the projection of the spin state of the electron on the axis along the external magnetic field. Figure 3 (c) illustrates a spin readout method based on spin-selective tunneling [16-18]. For this, a readout point is found, where spin-down and spin-up states of the electron in the right dot lie below and above the reservoir chemical potential, respectively. Pulsing to this position when the electron is in spin-up state, the electron will tunnel out into the reservoir, and a spin-down electron will tunnel back in its place. However, if the electron is in the spin-down state to begin with, it will stay confined in the dot, and no tunnel events will occur. The dot-reservoir tunnel rate is tuned into a suitable regime to be able to resolve the tunnel events within the bandwidth of our SET conductance measurement. Monitoring the SET conductance at a rate of one datapoint per microsecond, the tunnel out / in event pair is detected as a blip in the conductance as a function of time. If the signal exceeds a predefined threshold (dashed orange lines, Fig. 3 (c)), we infer that the electron tunneled out of the dot, indicating it was in the spin-up state. Since at the end of the readout, the state of the electron in the quantum dot is necessarily spin-down, the readout point also serves as the initialization point for the spin qubit.

In summary, this triad of initialization, manipulation and readout forms the backbone of our processing unit. More detailed operation sequences and functional flows will be illustrated in section 4.

3.3 Towards predictive qubit device models based on a Poisson/Schrödinger solver

Implementing the device manufacturing needs described in subsection 3.1 in shared R&D facilities is found to be a challenging task. Especially in view of scaling a quantum processing unit assistance of predictive simulations can become an essential component to expedite the design to qualification lifecycle. Therefore, we are currently developing a Poisson/Schrödinger solver with which the electrostatics of our spin qubit devices can be modelled.

In this approach the electrostatic potential of the qubit device follows from the Poisson equation. The charge density used in the Poisson equation can in principle be obtained from the wave function of the electrons. The wave function and the eigen energies are the solution of the Schrödinger equation, where the Hamiltonian for the Schrödinger equation contains a term with the electrostatic potential. This leads to a nonlinear coupling between the two equations. The workflow for solving the coupled equations is given in Fig. 4 (a). The Schrödinger equation can be solved with an initial value for the electrostatic potential, giving the wavefunctions to be used to calculate the charge density. The obtained charge density is then used to solve the Poisson equation, giving a new value for the electrostatic potential. This process is repeated until the difference in the electrostatic potential between two consecutive iterations becomes smaller than some defined tolerance value.

Simulation results for the spin qubit device are shown in Fig. 4 (b) and (c), based on the Thomas-Fermi approximation to calculate the charge density. The two figures display the simulated electron number density (#/nm^3) at the 2DEG area in the spin qubit device for specific values of the left and right plunger gate voltages; showing the double dot underneath the left and right plunger, tunnel coupling to the sensing dot, and the source and drain. The deviations in electron density in Figures 4 (b) and (c) result from differences in the geometry. Fig. 4 (b) is based on the designed geometry, while Fig. 4 (c) is based on the actual geometry derived from a scanning electron micrograph of the spin qubit device. From these two

figures we can see that charge occupation for the actual geometry is slightly deviating from the one for the designed geometry. Such a difference can lead to differences in the required voltage regime for qubit operations (not shown).

After improving the accuracy of these simulations, they can be helpful in speeding up qubit device design, making them potentially more robust against actual noisy fabrication. Hence, when fully operational this solver should serve as add-on onto classical process window enhancement techniques and therefore contribute to the design of the quantum processor. We envision that such solvers will become indispensable tools in the future for obtaining novel parameters such as tunnel coupling or charging energy which need to be seen as complementary to traditional lithographic ones such as edge placement error or line width roughness.

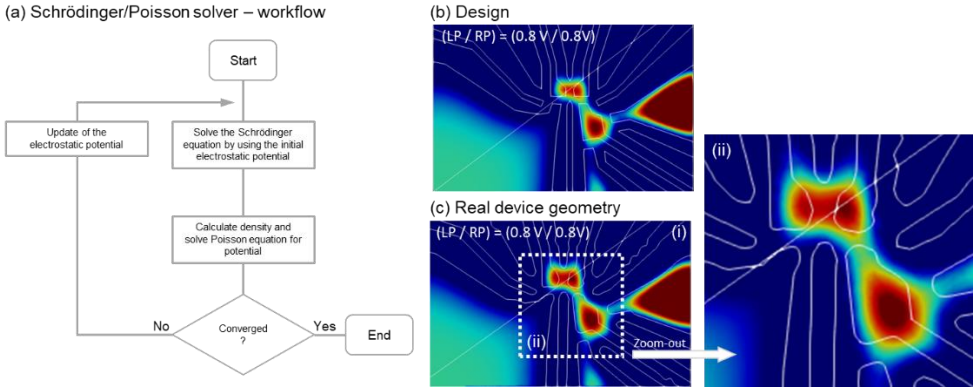

Figure 4: (a) The workflow for solving the coupled Schrödinger/Poisson equations is given in Fig. 4 (a). Simulation results for the spin qubit device are shown in Fig. 4 (b) and (c). (b) is based on the designed ideal geometry, while (c) is based on the actual geometry derived from a scanning electron micrograph of the spin qubit device. The two figures display the simulated electron number density at the 2DEG area in the spin qubit device for specific values of the left and right plunger gate voltages.

4. QUBIT CALIBRATION AND THE MAKING OF A QUANTUM GATE

A programmable quantum computer requires tune up and qubit calibration routines to match the quantum processing unit to the underlying high-level computing model, the circuit model of quantum computation [6, 20, 21]. Therefore in this section a few aspects will be shown of how the quantum processing unit will be made ready for computing. In subsection 4.1 we will return to the control and manipulation sequences, but with a bird's eye view in mind, describing the steps needed between executing an algorithm and receiving the result of this computation on the QI web interface. Subsection 4.2 will illustrate QI's calibration framework such that qubits can be appropriately tuned up for information processing and in subsection 4.3 a concrete matching of an experiment to a component of the circuit model, namely the Pauli X gate, will be presented.

4.1 Qubit operation sequences

Quantum algorithms consist of a series of commands acting on qubits, just like a classical program consists of commands acting on bits. As illustrated in section 3, every spin qubit device is probed by a sequence consisting of initialization-manipulation-readout. The exact scheme depends on the specifics of the device. In this section we will look at the control and manipulation from a higher-level viewpoint. The functional flow of the operation is shown in Figure 5 (i) through (viii). Note this basic operation sequence is repeated many times (typical number: 1024) in order to collect sufficient measurement statistics, taking into account the measurement error. As an example, we will look at a simple Bell state experiment. In this experiment two qubits are entangled using a Hadamard gate (H) and a controlled rotation gate (CNOT). In Figure 5 (i) the experiment is defined in the Quantum Inspire interface. When a user submits this algorithm to the QI system, the following steps are performed. First the transpiler decomposes the H gate into two elementary gates: an X180 and a mY90 gate. The reason for this is that our qubit device does not support the requested Hadamard gate directly. Also the CNOT gate is decomposed into more elementary gates. At this stage the quantum compiler can also perform optimizations, for example combining two X gates into a single identity operation, or changing the order of commuting

operations. The elementary gates of our system (several single-qubit rotations and a CZ gate) allow any 2-qubit gate to be expressed (they form a universal set [6]). Next the low-level compiler translates the elementary gates to DC pulses and microwave pulses. Single qubit rotations are performed using microwave pulses at the qubit resonance frequency [14, 16-19]. The controlled phase gate is performed by changing the exchange coupling between the two qubits using a short pulse on a combination of gates on the sample [14, 16-19]. The microwave pulses are generated using upconversion of an IQ pair generated by an arbitrary waveform generator using a vector source. Given the input signals I(t) and Q(t) generated by the AWG, the output M(t) of the vector source is given by $\mathrm{M}(t) = \mathrm{I}(t)\, sin(2\pi\varpi t) + \mathrm{Q}(t)\cos(2\pi\varpi t)$. By carefully defining I and Q, we can generate any signal we need. The upconversion allows us to combine the flexibility of the AWG together with the quality of the vector source local oscillator at the high frequency ω. We generate the waveforms dynamically based on the quantum gate parameters. The dynamic generation allows us to use software reference frames that allow us to optimize away all Z rotations. A simplified version of the waveforms generated for the Bell experiment is depicted in Figure 5 (iii). In reality the durations are different: Manipulation is in the nanosecond scale, the initialization and readout on microsecond scale. The generated microwave and pulse signals are sent to the device using the electronics (iv, v). The raw signal of the charge sensor is acquired and converted into spin-up or spin-down counts (more details on this are to be found in section 3). The results are either displayed on the website (Figure 5 (viii)) or processed further for other applications.

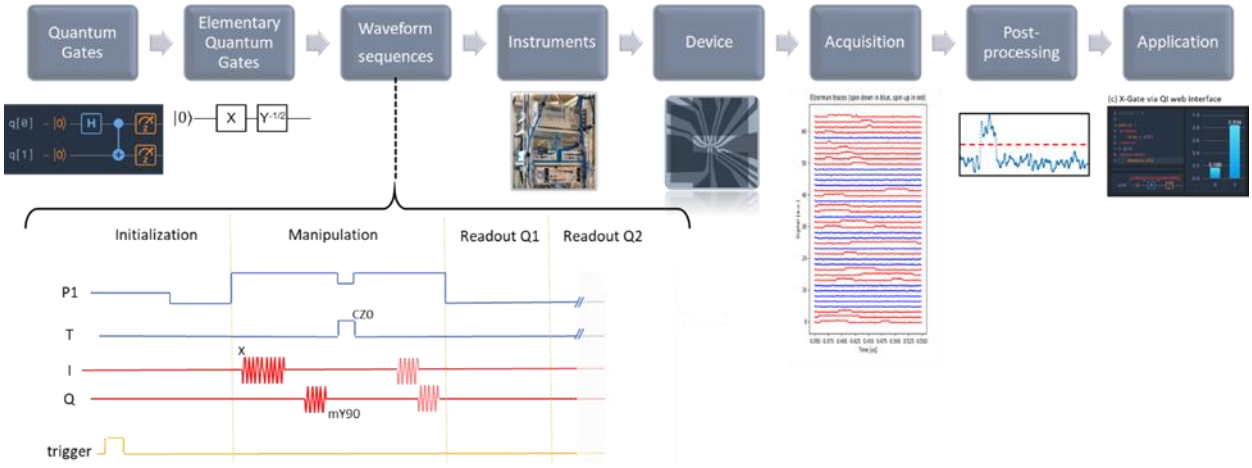

Figure 5: Full system walkthrough: i) Bell state experiment; ii) Decomposition into elementary gates; iii) Simplified representation of waveforms; iv) Instrument commands; v) Pulses to device, amplification of signal; vi) Acquisition of raw signal; vii) Processing to fractions of spin-up and down; viii) Result available.

4.2 The calibration framework

The quantum processing devices need to be calibrated. This involves both the bootstrapping (a series of experiments designed to arrive at the point where we have proper operation) and tracking of parameters over time due to drift in the device. The bootstrapping phase involves elementary checks such as checking connectivity of the cables, but also more complex tasks such as choosing the magnetic field strength for operation. During bootstrapping all parameters required for operation of the system are determined. For example: the Pauli X gate consists of a single microwave pulse. The duration and power need to be calibrated in order to achieve good fidelity. For even higher fidelities more parameters such as the pulse shape, or gate latencies need to be determined. In this section we describe the calibration framework that is used to calibrate the device. One of the main advantages of using the structure is that it helps keeping the different components of the system modular. Even where components are connected, the structure makes this very explicit. The calibration framework allows both structured (human-assisted) and automated calibration of the device. The center of the framework is the CalibrationGraph [20, 21] shown in Figure 6. This CalibrationGraph contains a list of (device specific) calibration methods and their relations. Each node of the graph corresponds to a calibration and the edges specify the dependencies. Each calibration node consists of one or more of the following steps. *Measure*: Perform a measurement on the device to extract information about the state of the system. From the raw measurement data the spin-up and spin-down fractions are determined. *Analyze*: The acquired data is analyzed to determine parameters such as the qubit resonance frequency, coherence time, etc. *Update*: Based on the results of the previous steps we update the system properties. Let us take as an

example the use of Rabi oscillations for the calibration of the duration and amplitude of a microwave pulse to implement an X gate. The *measure* component consists of a 1D experiment where an X pulse is applied with variable duration. The waveforms corresponding to this experiment are generated and uploaded to the AWG and the spin-up and -down fractions are recorded. The *analysis* consists of fitting a damped sine wave to the fractions, from which the Rabi frequency and visibility can be determined. The *update* step consists of updating the definition of the X gate duration and amplitude based on the calibrated Rabi frequency. The example just described corresponds to the nodes "time Rabi" (*measure* and *analyze*) and "update_rabi_frequency" (*update*) in the calibration graph. In Figure 6 one can see the dependencies of the "time Rabi" node. Before a time Rabi can be performed the readout needs to be calibrated ("select_readout_point") and the qubit frequency needs to be calibrated ("update_qubit_frequency" node). The CalibrationGraph is executed periodically and checks whether any node of the graph needs to be maintained. If a node is maintained, then it is checked whether the status of the node is good or not. If not, then first all subnodes of the node are maintained. If all subnodes have status good, then the calibration corresponding to the current node is executed. The logic flow of the calibration graph is described in [20, 21].

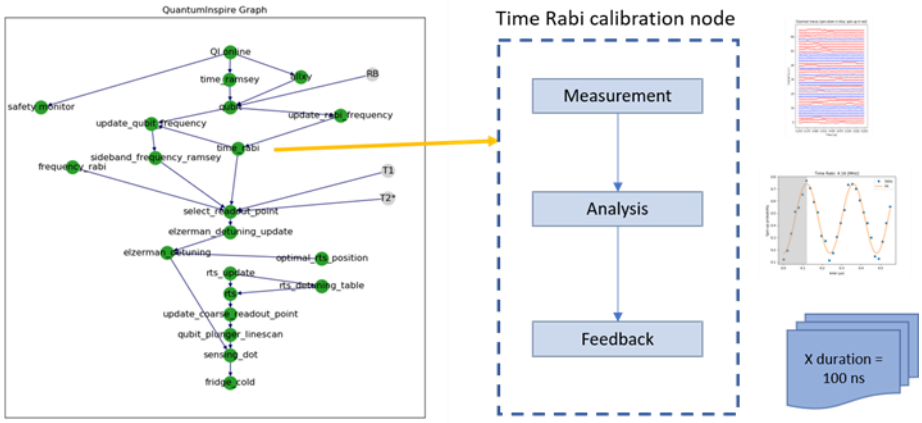

Figure 6: Calibration framework. a) The CalibrationGraph, b) Time Rabi node of the graph c) Stages of the node. The measurement consists of a 1D experiment where the duration of the X pulse is varied. In the analysis the Rabi frequency is determined by fitting a damped since wave to the data. In the feedback stage the duration of the X pulse is updated in the database.

4.3 The Pauli X gate

One of the most basic operations to a qubit is a single rotation of 180 degrees around the x-axis. This operation is called the Pauli X gate. If the qubit starts in the $|0\rangle$ state, then after applying the X gate the qubit will be in the $|1\rangle$ state. We can perform a qubit rotation by applying a microwave pulse with certain frequency and amplitude. A good tool to learn about the behavior of the qubit is to measure a Chevron pattern (Figure 7 (a)). Here we perform a series of measurements where we vary two parameters: the frequency of the pulse and the duration of the pulse (the other parameters such as amplitude and phase are fixed). For each combination of frequency and pulse duration we initialize the qubit, apply the microwave pulse and measure the qubit state. Initially the qubit state is in the ground state $|0\rangle$, which is represented by the dark band in Figure 7 (a) for duration zero. For longer duration of the pulse, the qubit rotates to state $|1\rangle$ (yellow color). The oscillations form a so-called Chevron pattern. The pattern is symmetric around the qubit resonance frequency (Larmor frequency), which is the preferred frequency of operation for the qubit. The Chevron pattern is mainly used during the bootstrapping phase. For automated calibrations we use 1D slices of the pattern which are faster to acquire. For example, to calibrate the Pauli X gate we perform a time Rabi experiment. This is a horizontal slice through the Chevron pattern at the qubit resonance frequency. For a typical Rabi calibration experiment the time duration is varied in a number of steps and for each step a number (1024) of single-shot measurements is performed. The results of a typical time Rabi experiment are shown in Figure 7 (b). For the X gate we take the shortest duration possible to rotate the qubit from the down to the up state. Here we find 200 ns. Our software does this by fitting a damped sine wave to the plot of Figure 7 (b) [18]. After calibration of the X gate we can define other gates in terms of the X gate parameters, such as the X90 gate and the Y gate (phase shift on the microwave pulse by 90 degrees). Applying an X gate from a user algorithm is a bit more involved. It requires the power (user set during bootstrapping), the frequency (determined in a frequency Ramsey calibration), the total

integral, e.g. duration and amplitude (duration is determined during bootstrapping, amplitude using a time Rabi). Finally, the phase of the microwave pulse needs to be determined. For the X gate the phase is 0 per definition. But in the software reference frame the phase of the actual microwave pulse depends on the history of gates applied before the current X gate.

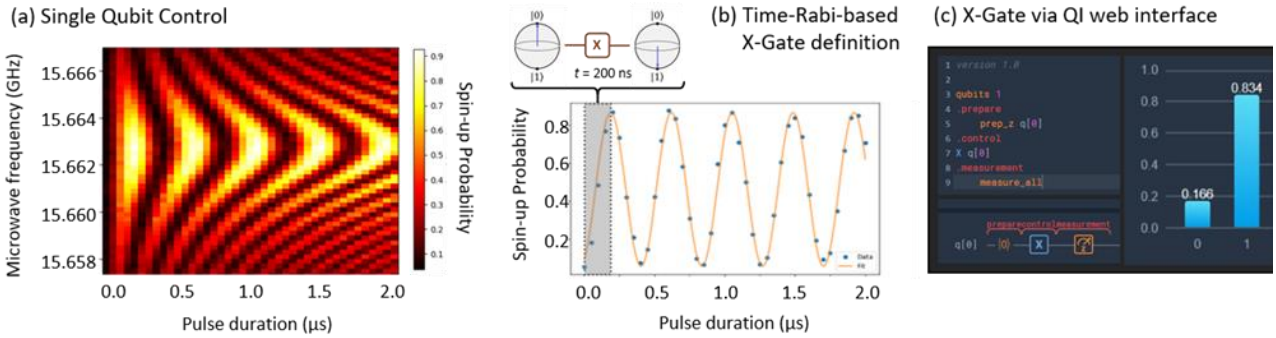

Figure 7: (a) Chevron patterns of a single qubit control measurement. (b) Time-Rabi-based X gate definition as example of a quantum gate. (c) Execution of the Pauli X gate on quantum inspire after calibration. Note that the spin-up fraction (0.834) corresponds to the amplitude of the first peak of the time Rabi experiment in (b). This indicates that the calibration was performed successfully.

Similar matching of experiments will ensure the availability of a universal set of single-qubit and two-qubit gates ready for quantum computation with Quantum Inspire.

5. SUMMARY & OUTLOOK

Quantum computing is now steadily maturing towards a novel discipline where full-stack system considerations and interactions therein are becoming central pillars for future research, development, and engineering in this field. QuTech in Delft, devoted to R&D in Quantum Technology, is therefore developing Quantum Inspire, its full-stack prototype platform for collaboration and co-development in quantum computing (www.quantum-inspire.com). A prerelease of QI is already offering the public access to QuTech's QX quantum computing simulator which can emulate a system with up to 31 qubits. Access to a CMOS-compatible Silicon spin qubit-based quantum processor will be provided in the future. Here we presented important aspects of this future hardware back-end, namely its manufacturing, electrical characterization and calibration to make it information processing-ready.

To conclude, full-stack systems are being built and made accessible to the broader public for explorative research, for educational purposes and also in part to prepare for the next generation of systems to come. They are still in their very early stages of development. Considerable scientific as well as engineering challenges are still looming and need to be tackled on the long road towards a universal quantum computing system capable of solving real-world problems. New physics is there to be discovered. Qubits still have personalities and are susceptible to errors. Electrical signals need to be sent and retrieved from each and every qubit. Hence, when the systems scale up a wiring bottleneck is looming. Power management, volume and scalability in conjunction with the available cryogenic environment needs be evaluated. Scalable control equipment and circuits need to be developed. Those are only but a few of a long list of challenges. Still, as already mentioned, full-stack systems are technology accelerators. They will help detecting those current and future gaps in the QC supply chain and innovation roadmap. Having full-stack systems at ones disposal will help working out what exactly quantum computing's place will be in a future computing ecosystem. QuTech in Delft is fully supporting these ongoing world-wide efforts with its quantum computing platform Quantum Inspire.

REFERENCES

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SIAM Journal of Computing 26, pp. 1484-1509 (1997).

[2] J. Preskill, "Quantum Computing in the NISQ era and beyond", arxiv:1801.00862v3 (2018).

[3] M. van de Kerkhof, H. Jaspers, L. Levasier, R. Peeters, R. van Es, J.-W. Bosker, A. Zdravkov, E. Lenderink, F. Evangelista, P. Broman, B. Bilski, T. Last, "Enabling sub-10nm node lithography: presenting the NXE:3400B EUV scanner", Proc. of SPIE 10143, Extreme Ultraviolet (EUV) Lithography VIII, 101430D (2017).

[4] X. Fu, L. Riesebos, L. Lao, C. G. Almudever, F. Sebastiano, R. Versluis, E. Charbon, K. Bertels, "A Heterogeneous Quantum Computer Architecture", Proceedings of the ACM International Conference on Computing Frontiers, 323–330 (ACM, 2016).

[5] N. C. Jones, R. van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, Y. Yamamoto, "Layered Architecture for Quantum Computing", Phys. Rev. X 2, 031007(27) (2012).

[6] M. Nielsen, I. Chuang, "Quantum Computation and Quantum Information", Cambridge UP (2000).

[7] F. Arute, …, H. Neven, J. M. Martinis, "Quantum supremacy using a programmable superconducting processor", Nature 574, 505-510 (2019).

[8] B. M. Terhal, "Quantum Error Correction for Quantum Memories", Rev. Mod. Phys 87, 307 (2015).

[9] C. K. Andersen, A. Remm. S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, A. Wallraff, "Repeated Quantum Error Detection in a Surface Code", arxiv:1912.09410v1 (2019).

[10] N. Khammassi, I. Ashraf, X. Fu, C. G. Almudever, K. Bertels, "QX: A High-Performance Quantum Computer Simulation Platform", DATE conference, 464-9 (2017).

[11] https://projectq.ch/

[12] https://qiskit.org/

[13] G. Popkin, "Quest for Qubits", Science 354, 6316, 1090-1093 (2016).

[14] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, M. Veldhorst, "Interfacing spin qubits in quantum dots and donors – hot, dense, and coherent", NOP Quantum Information 3:34 (2017).

[15] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, W. D. Oliver, "Superconducting Qubits: Current State of Play", Annual Review of Condensed Matter, Physics Vol. 11 (2020).

[16] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, L. M. K. Vandersypen, "Spins in few-electron quantum dots", Rev. Mod. Phys. 79, 1217(49) (2007).

[17] F. A. Zwanenburg, A. S. Dzurak, L. C. L. Hollenberg, G. Klimeck, S. Rogge. S. N. Coppersmith, M. A. Eriksson, "Silicon quantum electronics", Rev. Mod. Phys. 85, 961(59) (2013).

[18] T. F. Watson, S. G. J. Philips, E. Kawakami, D. R. Ward, P. Scarlino, M. Veldhorst, D.E. Savage, M. G. Lagally, M. Friesen, S. N. Coppersmith, M. A. Eriksson, L. M. K. Vandersypen, "A programmable two-qubit quantum processor in silicon", Nature 555, 633-637 (2018).

[19] L. M. K. Vandersypen, "1.4 Quantum Computing – The next challenge in circuit and system design", 2017 IEEE International Solid-State Circuits Conference (ISSCC), 24-29 (2017).

[20] J. Kelly, P. O'Malley, M. Neeley, H. Neven, J. M. Martinis, "Physical qubit calibration on a directed acyclic graph", arxiv:1803.03226v1 (2018).

[21] P. T. Eendebak, et al. Quantum Technology Toolbox; available from: https://github.com/qutech-delft/qtt (2019).