SpinQ Triangulum

A commercial three-qubit desktop quantum computer.

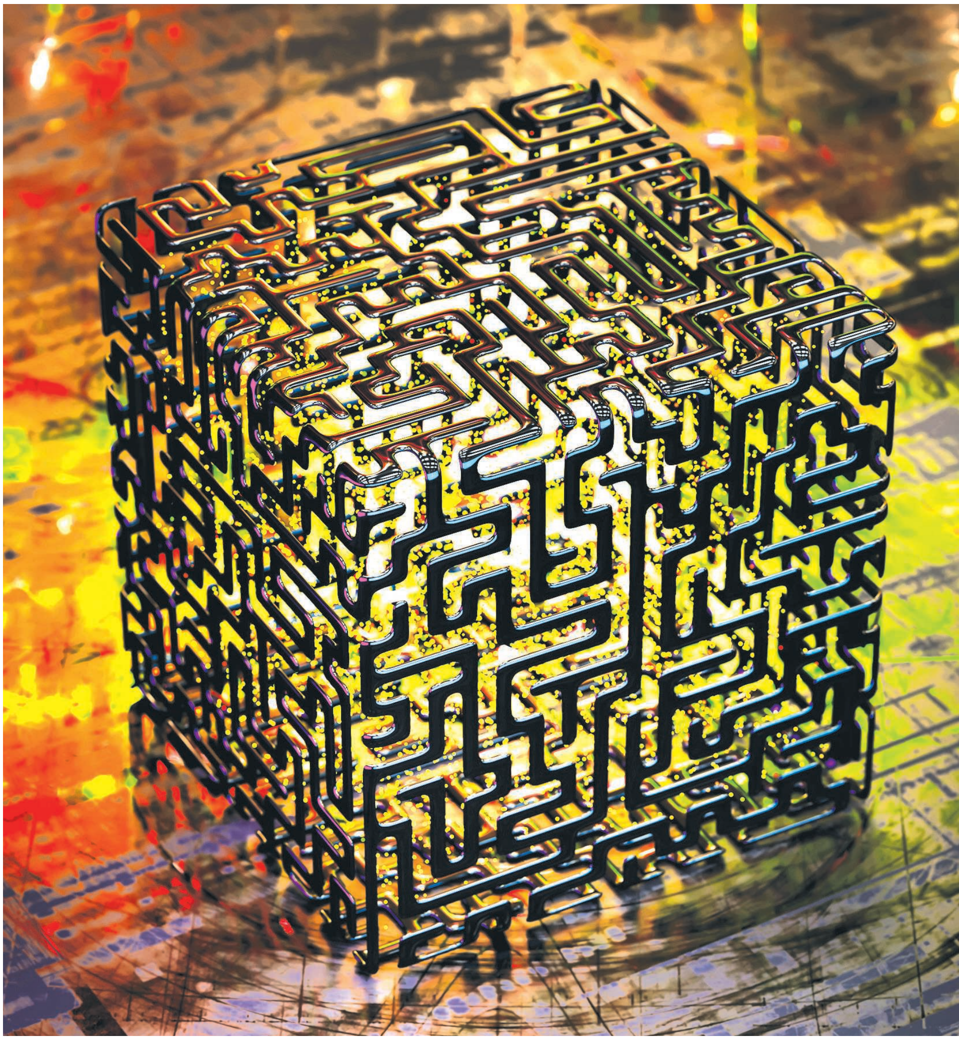
SPINQ TRIANGULUM IS THE SECond generation of the desktop quantum computers designed and manufactured by SpinQ Technology. SpinQ's desktop quantum computer series, based on a room-temperature nuclear magnetic resonance (NMR) spectrometer, provides lightweight, cost-effective, and maintenance-free quantum computing platforms that aim to provide real-device experience for quantum computing education for kindergarten through 12th grade (K–12) and the college level. These platforms also feature quantum control design capabilities for studying quantum control and quantum noise.

Compared with the first-generation product, the two-qubit SpinQ Gemini, Triangulum features a three-qubit quantum processing unit (QPU), smaller dimensions ($61 \times 33 \times 56$ cm^3), and a lighter weight (40 kg). Furthermore, the magnetic field is more stable, and the performance of the quantum control is more accurate.

This article introduces the system design of Triangulum and its new features. As an example of performing quantum computing tasks, we present the implementation of the Harrow–Hassidim–Lloyd (HHL) algorithm on Triangulum, demonstrating Triangulum's capabilities for undertaking complex quantum computing tasks. SpinQ will continue to develop desktop quantum computing platforms with more qubits. Meanwhile, a simplified version of SpinQ Gemini, namely, Gemini Mini, has been recently realized. Gemini Mini is much more portable (at $20 \times 35 \times 26$ cm^3 and 14 kg) and affordable for most K–12 schools around the world.

BACKGROUND

SpinQ Triangulum is the second-generation product of SpinQ's commercial



©SHUTTERSTOCK.COM/PLOTPLOT

GUANRU FENG, SHI-YAO HOU, HONGYANG ZOU, WEI SHI, SHENG YU, ZIKAI SHENG, XIN RAO, KAIHONG MA, CHENXING CHEN, BING REN, GUOXING MIAO, JINGEN XIANG, AND BEI ZENG

Digital Object Identifier 10.1109/MNANO.2022.3175392
Date of current version: 7 June 2022

1932-4510/22©2022IEEE

desktop quantum computing platform series [1]–[3]. Similar to the first-generation product, the two-qubit SpinQ Gemini, Triangulum is based on an NMR system, which was among the very first systems developed for quantum computing and has developed many advanced quantum control techniques [4]–[30]. Compared with Gemini, Triangulum accommodates a three-qubit QPU. The weight and size are further reduced to 40 kg and $61 \times 33 \times 56\ \mathrm{cm}^3$, respectively. Figure 1 shows the exterior look of Triangulum and its user-friendly interface SpinQuasar, where users can compose quantum circuits and interact with the QPU.

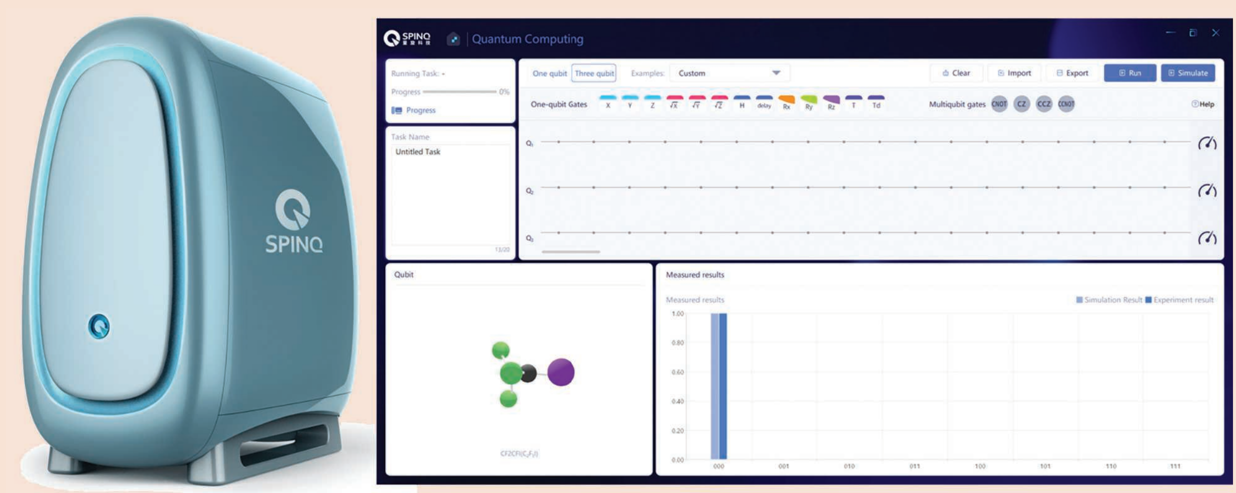
SpinQ desktop quantum computing platforms take advantage of the recent development of small permanent-magnet technology to reach a small size and weight.

Most of the quantum computers in research labs are out of reach in real life due to their cost, weight, volume, and extreme physical conditions, such as traditional NMR quantum computing, which is performed on huge and expensive commercial superconducting NMR spectrometers, and superconducting qubit quantum computing, which requires an extreme-temperature environment. SpinQ desktop quantum computing platforms take advantage of the recent development of small permanent-magnet technology [31] to reach a small size and weight. SpinQ Gemini is only 55 kg, and Triangulum is only 40 kg. Furthermore, compared with their counterparts in research labs, SpinQ Gemini and Triangulum are cost-effective and require no special maintenance; hence, they are very friendly to everyone who is interested in quantum computing.

SpinQ Triangulum inherits Gemini's powerful functions of quantum algorithm circuit design and programming using its SpinQuasar software (Figure 1) as well as the demonstrations of 10+ famous quantum algorithms, such as the Deutsch algorithm [32] and Grover algorithm [33], [34]. Thus, it provides a very friendly platform for nonspecialists who aim to learn quantum computing basics and quantum programming. Furthermore, Triangulum has improved its stability and quantum control accuracy. Its powerful arbitrary-waveform-generation function enables advanced control of the quantum system. With its three-qubit QPU, Triangulum can serve as a powerful tool for quantum computing-related research under real-world conditions.

SYSTEM

The overall schematic diagram is shown in Figure 2. Similar to Gemini, Triangulum is composed of a PC with SpinQuasar, a control system on the master board, a temperature-control module, a pair of permanent magnets, a field-shimming system, a radio-frequency (RF) system, and a tube of sample. Additionally, Triangulum has a field-locking system, and its pulse generator enables an arbitrary-waveform-generation function.

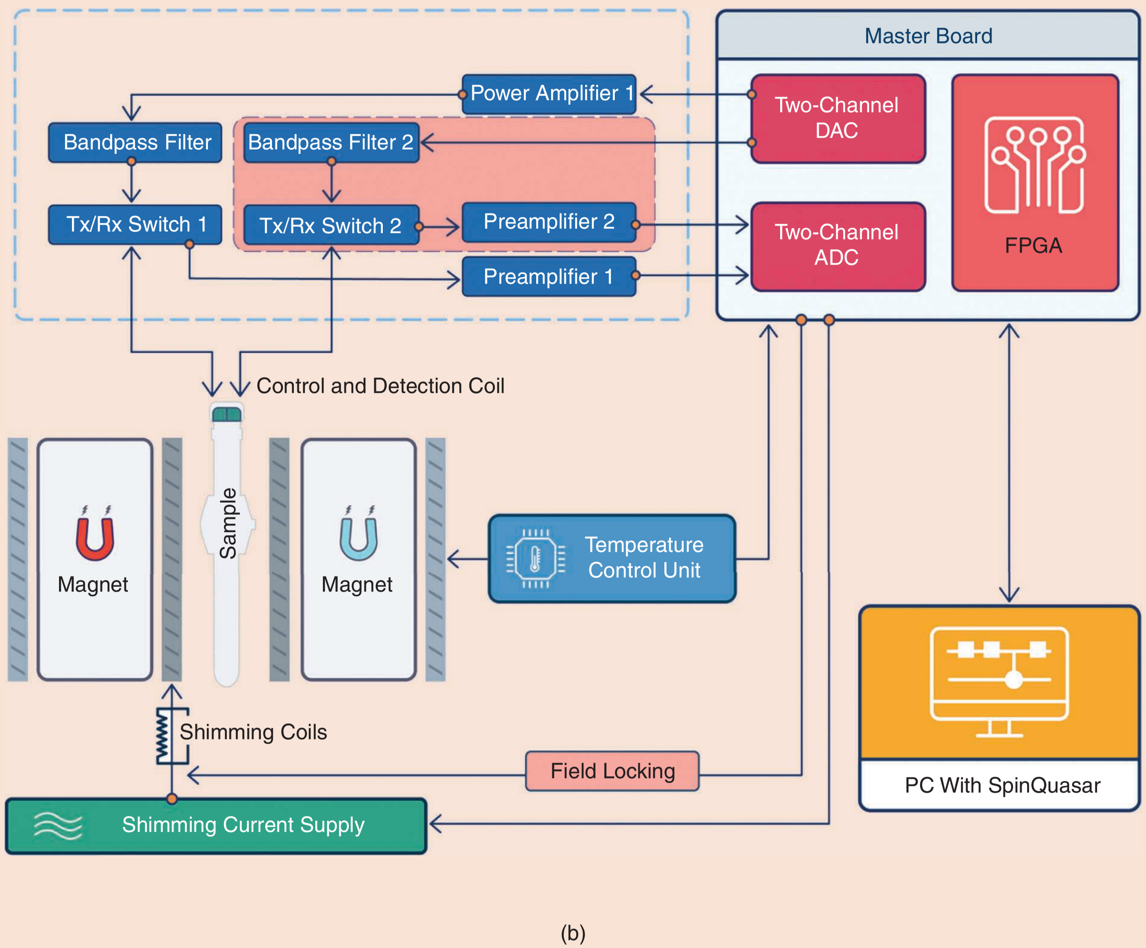


(a) (b)

FIGURE 1 (a) The Triangulum and (b) user-friendly interface SpinQuasar [1]. SpinQuasar is installed on a PC that connects with Triangulum. Users can manipulate Triangulum conveniently via SpinQuasar. On this particular page, users can compose quantum circuits, implement quantum algorithms, and check the computation results. There are two buttons, "Run" and "Simulate," for activation of the experiment and the simulation, respectively. The probabilities of the eight eigen bases from the experiment and the simulation are shown in the bottom half of the interface.

	Gemini	Triangulum
QPU	Two Qubits	Three Qubits
SpinQuasar	Yes	Yes
Cloud Service	Yes	Yes
API	Yes	Yes
Weight	55 kg*	40 kg
Dimensions	$70\times40\times80^{*}$ cm^3	$61\times33\times56$ cm^3
RF Channels	2	2
Arbitrary Waveform Generation	No	Yes
Field Shimming	Yes	Yes
Temperature Control	Yes	Yes
Field Locking	No	Yes
Measurement Speed	~3 min	~55 s

(a)



(b)

FIGURE 2 (a) A comparison between Gemini and Triangulum. *These are the data of the first version of Gemini. The weight and dimensions of the current version of Gemini are similar to those of Triangulum. (b) An overview of the schematic diagram of the Triangulum system. The master board, equipped with an FPGA, provides the control logic of Triangulum. SpinQuasar communicates with the FPGA through USB so that the user can access Triangulum. The magnets, together with the temperature-control unit, the field-shimming system, and the field-locking system, provide a stable static homogeneous magnetic field. The RF module amplifies the RF control pulses and detects the RF signals from the qubits. The field-locking system utilizes the RF signal route that is composed of bandpass filter 2, Tx/Rx switch 2, and preamplifier 2, which are in the orange box. ADC: analog-to-digital converter; API: application programming interface; DAC: digital-to-analog converter; FPGA: field-programmable gate array; RF: radio frequency; Rx: receiver; Tx: transmitter.

The modules that Triangulum inherits from Gemini realize most of the same functions [3]. The master board includes a field-programmable gate array (FPGA), an analog-to-digital converter, and a digital-to-analog converter, which together realize the algorithms required for pulse generation, signal processing, and so on. Triangulum's pulse-generation function is more powerful than Gemini's and has an arbitrary-waveform-generation ability, which is discussed later. SpinQuasar is the software interface for users, and it provides an interface to the QPU as well as the instrument calibration. Advanced functions, such as cloud computing [35] and application programming interfaces for programmable control, are also supported by Triangulum.

The NMR sample used in Triangulum is iodotrifluoroethylene (C_2F_3I). Different from Gemini, which uses two different types of nuclei as qubits, Triangulum uses three ^{19}F nuclei as three qubits. On one hand, only one RF channel is needed for the excitation and signal detection for the three qubits, which seems to be a great advantage. On the other hand, the simple square pulses used on Gemini are not enough anymore to manipulate nuclei of the same type as different qubits. Pulses with arbitrary shapes, such as the gradient ascent pulse engineering (GRAPE) pulses [27], are needed for accurate quantum control. Therefore, the control of three qubits of the same type of spin requires the pulse generator and the RF amplification/transmission system to be more powerful. In Triangulum, the pulse-generation module of the FPGA is developed to be capable of arbitrary-waveform generation, with a magnitude accuracy of 1/65,536 and phase accuracy of $2\pi/65{,}536$. The RF system, which is responsible for the pulse amplification and transmission, is also improved to faithfully transmit arbitrary waveforms.

To realize the stable and homogeneous magnetic field required by quantum computing, Gemini uses a pair of NdFeB plate permanent magnets, a field-shimming system, and a temperature-control system. In addition to those modules, Triangulum is equipped with a field-locking system to make the static magnetic field more stable. The increased field stability satisfies the requirement of specially designed pulse shapes that are very sensitive to the resonance frequency fluctuations of qubits and, thus, very sensitive to the field fluctuation. Locking is realized by continuously exciting and detecting ^{1}H signals of acetone. By analyzing the detected ^{1}H frequency, the field drift can be estimated. Then, a compensation field is generated by coils to make the magnet field stable at a desired magnitude. The ^{1}H spin excitation and signal detection are realized using the second RF channel, as shown in Figure 2. Therefore, with the additional abilities—field locking and arbitrary-waveform generation—the three-qubit system of Triangulum can be accurately manipulated for different quantum computing tasks.

> Triangulum has a field-locking system, and its pulse generator enables an arbitrary-waveform-generation function.

QUANTUM COMPUTATION

SOFTWARE INTERFACE

Like Gemini, SpinQuasar is provided to users as the interface to Triangulum (Figure 1). Most parts of the interface are the same as the version on Gemini. The quantum circuits as well as the available quantum gates are shown in the top half of the interface. A dropdown list provides users with built-in algorithms. The "Run" and "Simulate" buttons on the right of the circuits can be used to initiate the experiments on the quantum processor or the embedded quantum simulator. The results are given at the bottom right. It should be mentioned that users can click on or off the measurement labels to the rightmost of the lines, which stand for qubits. The experiment or simulation will only give the measurement results for the qubits that users choose. In the following section, we introduce the molecule system of the quantum processor, available quantum gates, initial state, and measurement of the quantum processor state.

THE SPIN SYSTEM

The sample we use is iodotrifluoroethylene (C_2F_3I). The molecules are placed in the center of parallel permanent magnets. A ^{19}F nucleus is a spin-half system. When placed in a static magnetic field, it has two energy levels; thus, a ^{19}F spin can be used as a qubit. The Lamor frequency of ^{19}F in a 1-T magnetic field is ~40 MHz. Its state can be manipulated by irradiating electromagnetic waves (pulses) with frequencies close to its Larmor frequency. The three ^{19}F nuclei in iodotrifluoroethylene are used as the three qubits. The structure and the parameters of the sample are listed in Figure 3. The T_1 and T_2 for the ^{19}F spins are about 7 and 0.2 s, respectively. The J couplings between the three spins are –128, 68, and 49 Hz. The frequencies of the three ^{19}F spins, located around 40 MHz, are slightly different. These differences are usually called *chemical shifts*. The excitation profile in the frequency domain of simple square pulses is usually broad. Because of the small frequency differences among the three spins, it is difficult to realize accurate individual controls using simple square pulses. Therefore, GRAPE pulses are used to control the three-qubit system. The spin Hamiltonian in the rotating frame is

$$
\begin{aligned}
H_0 = 2\pi(\nu_1 I_z^1 + \nu_2 I_z^2 + \nu_3 I_z^3) \\
+ \sum_{k=x,y,z} 2\pi(J_{1,2} I_k^1 I_k^2 + J_{1,3} I_k^1 I_k^3 \\
+ J_{2,3} I_k^2 I_k^3),
\end{aligned} \tag{1}
$$

where ν_i values are chemical shifts for the ith nuclei, and $J_{i,j}$ values are J couplings between the ith and jth nuclei.

THE GATE SET

The global 90° rotation of the three qubits can be realized using 10-μs square pulses. However, global control is not enough in quantum computing. As mentioned, on Triangulum, GRAPE pulses are used to realize all of the quantum gates available. The available quantum gates contain single-qubit, two-qubit, and three-qubit gates. The single-qubit gates are Pauli gates $(\sigma_x, \sigma_y, \sigma_z)$; 90° rotation gates; arbitrary rotation gates along the x-, y-, and z-axes; Hadamard gates; T gates; and the inverse of T gates. The two-qubit gates are controlled-Not (CNOT) and CZ gates between any pairs of the three qubits. The three-qubit gates are the Toffli gates with any two of the three qubits as the control qubits and the controlled-controlled-Not gates, which implement a π phase change to $|111\rangle$. It should be mentioned that all of the gates are realized using a single GRAPE pulse, except the following cases: 1) all z rotation gates, including σ_z, 90° rotation, and arbitrary angle rotation gates along the z-axis, and 2) arbitrary rotation gates along the x- and y-axes.

In the first case, z rotation gates are realized virtually by changing the phase of the reference rotating frame [36]. To illustrate how this works, we first consider a simple square rotation pulse with rotating axis ϕ:

$$U = e^{-it\Omega(\cos\phi\sigma_x/2+\sin\phi\sigma_y/2)}, \tag{2}$$

where Ω and ϕ are the pulse amplitude and phase, respectively, which can be adjusted by the arbitrary-waveform generator. For any experiment, a reference phase ϕ_0 is set for observation so that all of the pulse phase is relative to ϕ_0:

$$U = e^{-it\Omega(\cos(\phi-\phi_0)\sigma_x/2+\sin(\phi-\phi_0)\sigma_y/2)}. \tag{3}$$

Changing $\phi-\phi_0$ means rotating the rotation axis in the xy-plane around the z-axis. If, within an experiment, we change the reference phase ϕ_0, it means the reference frame is rotated along the z-axis. For a given quantum state, this reference frame change is equivalent to a z rotation. Thus, by changing the reference frame of the pulses, which can be done conveniently in the observation, one can realize virtual z rotations. Furthermore, within an experiment, after any virtual z rotations, since the reference frame is changed, the ϕ of the subsequent rotations should be adjusted accordingly for the desired rotation axis. Here, we use the following sequence as an example:

$$R_x(\theta) - R_z\left(\frac{\pi}{2}\right) - R_x(\gamma). \tag{4}$$

We rotate the reference frame by $-\pi/2$ to realize $R_z(\pi/2)$ virtually. This means we need to change the observing frame by $-\pi/2$, and the sequence that needs to be implemented is

$$R_x(\theta) - R_{-y}(\gamma). \tag{5}$$

The $R_x(\gamma)$ after the virtual z rotation is changed, accordingly, to $R_{-y}(\gamma)$.

The gates in the second case, the arbitrary x or y rotation gates, are realized by first rotating the x- or y-axes to z using GRAPE pulses; then implementing a z rotation; and, finally, rotating the axes back to x or y.

THE PSEUDOPURE STATE

The initial state of the three-qubit system is prepared to be a pseudopure state (PPS) [4] using the same method as in Gemini. A PPS has the following form:

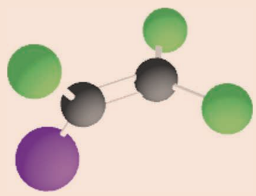
$$\rho_{\text{pps}} = \frac{1-\eta}{2^n} I^{\otimes n} + \eta |\psi\rangle\langle\psi|, \tag{6}$$

where $|\psi\rangle$ is a pure state. The PPS in (6) has the same unitary dynamics and observable effects as the pure state $|\psi\rangle$ except for the factor η. The PPS is widely used in NMR quantum computation.

Triangulum utilizes the relaxation method in [37] to prepare the three-qubit PPS starting from the thermal equilibrium state, whose state population is subject to a Boltzmann distribution at room temperature. The relaxation method uses repetitions of basis permutation operations and T_1 relaxation. Different from Gemini, whose permutation operation is realized by square pulses and delays, on Triangulum the basis permutation gate,

$$U_{\text{permute}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{7}$$

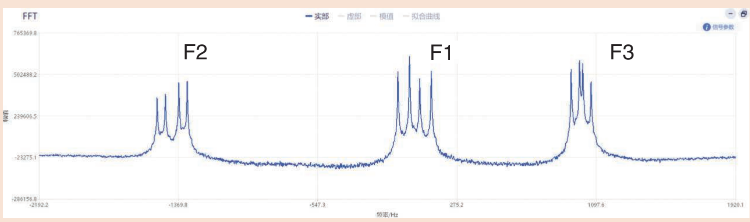
is realized by a GRAPE pulse (see Figure 4). U_{permute} permutes the state population between bases $|001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$, and $|111\rangle$ while leaving $|000\rangle$



(a)

	F1	F2	F3
F1	29.75	–128	68.2
F2	–128	–1,383.8	48.8
F3	68.2	48.8	999.5

(b)



(c)

FIGURE 3 (a) The molecule structure, (b) its parameter table, and (c) the Fourier transform spectrum of ^{19}F. In the molecule, there are three ^{19}F nuclear spins (green). The diagonal elements of the table are the chemical shifts of the three ^{19}F. The off-diagonal elements are their J couplings. All of the values in the table are in units of hertz. Each of the ^{19}F spins has four peaks in the Fourier transform spectrum, with peak splittings determined by the J coupling constants.

unchanged. This permutation operation is combined with a delay t after it. The permutation and the T_1 relaxation in the delay take effect alternately. After a certain number of cycles (N cycles), the system can reach a state whose dominantly occupied basis is $|000\rangle$, and the other seven bases have the same but smaller probability. This obtained state is a PPS and can be used as the initial state $|000\rangle$.

MEASUREMENTS

Different from Gemini, Triangulum only measures the diagonal elements of the system's density matrix after a certain gate sequence is applied. This means only probabilities of $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$, and $|111\rangle$ are measured. In NMR quantum computing, results are collected by ensemble measurements, which means one collects the ensemble expectation values of a certain operator. Specifically, NMR collects the $\langle \sigma_x + i\sigma_y \rangle$ of a certain qubit. To obtain the diagonal elements of the system's density matrix, one needs to measure $\langle \sigma_z^1 \rangle$, $\langle \sigma_z^2 \rangle, \langle \sigma_z^3 \rangle, \langle \sigma_z^1\sigma_z^2 \rangle, \langle \sigma_z^1\sigma_z^3 \rangle, \langle \sigma_z^2\sigma_z^3 \rangle$, and $\langle \sigma_z^1\sigma_z^2\sigma_z^3 \rangle$. The following expressions illustrate how the eight diagonal elements of the density matrix are calculated from those expectation values:

$$\rho_{11} = \frac{1}{8}(1+\langle\sigma_z^1\rangle+\langle\sigma_z^2\rangle+\langle\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\rangle +\langle\sigma_z^1\sigma_z^3\rangle+\langle\sigma_z^2\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{8}$$

$$\rho_{22} = \frac{1}{8}(1+\langle\sigma_z^1\rangle+\langle\sigma_z^2\rangle-\langle\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\rangle -\langle\sigma_z^1\sigma_z^3\rangle-\langle\sigma_z^2\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{9}$$

$$\rho_{33} = \frac{1}{8}(1+\langle\sigma_z^1\rangle-\langle\sigma_z^2\rangle+\langle\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\rangle +\langle\sigma_z^1\sigma_z^3\rangle-\langle\sigma_z^2\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{10}$$

$$\rho_{44} = \frac{1}{8}(1+\langle\sigma_z^1\rangle-\langle\sigma_z^2\rangle-\langle\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\rangle -\langle\sigma_z^1\sigma_z^3\rangle+\langle\sigma_z^2\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{11}$$

$$\rho_{55} = \frac{1}{8}(1-\langle\sigma_z^1\rangle+\langle\sigma_z^2\rangle+\langle\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\rangle -\langle\sigma_z^1\sigma_z^3\rangle+\langle\sigma_z^2\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{12}$$

$$\rho_{66} = \frac{1}{8}(1-\langle\sigma_z^1\rangle+\langle\sigma_z^2\rangle-\langle\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\rangle +\langle\sigma_z^1\sigma_z^3\rangle-\langle\sigma_z^2\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{13}$$

$$\rho_{77} = \frac{1}{8}(1-\langle\sigma_z^1\rangle-\langle\sigma_z^2\rangle+\langle\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\rangle -\langle\sigma_z^1\sigma_z^3\rangle-\langle\sigma_z^2\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle), \tag{14}$$

$$\rho_{88} = \frac{1}{8}(1-\langle\sigma_z^1\rangle-\langle\sigma_z^2\rangle-\langle\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\rangle +\langle\sigma_z^1\sigma_z^3\rangle+\langle\sigma_z^2\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle). \tag{15}$$

It should be noted that all of the $\langle \sigma_z^1 \rangle$, $\langle \sigma_z^2 \rangle, \langle \sigma_z^3 \rangle, \langle \sigma_z^1\sigma_z^2 \rangle, \langle \sigma_z^1\sigma_z^3 \rangle, \langle \sigma_z^2\sigma_z^3 \rangle$, and $\langle \sigma_z^1\sigma_z^2\sigma_z^3 \rangle$ values are not direct observable. Additional readout pulses are needed to transform them to be observable. A total of three experiments are needed to obtain the seven values. Specifically, in the three experiments, each of the three qubits is measured after a 90° readout pulse on that particular qubit. In each experiment, four peaks of the measured qubit are obtained in the spectrum whose amplitudes are linear combinations of $\langle \sigma_z^1 \rangle$, $\langle \sigma_z^2 \rangle, \langle \sigma_z^3 \rangle, \langle \sigma_z^1\sigma_z^2 \rangle, \langle \sigma_z^1\sigma_z^3 \rangle, \langle \sigma_z^2\sigma_z^3 \rangle$, and $\langle \sigma_z^1\sigma_z^2\sigma_z^3 \rangle$. For example, in one of the three experiments, the first qubit is observed after an $R_y^1(\pi/2)$ rotation. Then, the real parts of the four peaks of the first qubit in the Fourier transform (FT) spectrum are proportional to the following values:

$$\begin{aligned}
&\langle\sigma_z^1\rangle-\langle\sigma_z^1\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\rangle-\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle,\\
&\langle\sigma_z^1\rangle+\langle\sigma_z^1\sigma_z^3\rangle+\langle\sigma_z^1\sigma_z^2\rangle+\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle,\\
&\langle\sigma_z^1\rangle-\langle\sigma_z^1\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\rangle+\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle,\\
&\langle\sigma_z^1\rangle+\langle\sigma_z^1\sigma_z^3\rangle-\langle\sigma_z^1\sigma_z^2\rangle-\langle\sigma_z^1\sigma_z^2\sigma_z^3\rangle.
\end{aligned} \tag{16}$$

By fitting the spectra and solving linear equations, $\langle \sigma_z^1 \rangle, \langle \sigma_z^2 \rangle, \langle \sigma_z^3 \rangle, \langle \sigma_z^1\sigma_z^2 \rangle$, $\langle \sigma_z^1\sigma_z^3 \rangle, \langle \sigma_z^2\sigma_z^3 \rangle$, and $\langle \sigma_z^1\sigma_z^2\sigma_z^3 \rangle$ can be obtained, and, thus, the ρ_{ii} values can be calculated for i from one to eight. It only takes ~55 s for Triangulum to reconstruct the diagonal elements of the density matrix, faster than Gemini, which needs six experiments for density matrix reconstruction.

APPLICATION: HHL ALGORITHM FOR LINEAR SYSTEMS OF EQUATIONS

In this section, we demonstrate the realization of the HHL algorithm for linear systems of equations on Triangulum. Solving linear systems of equations is a problem present in almost all areas of science and engineering. The HHL algorithm [38] is a quantum algorithm for solving linear systems of equations. Under certain conditions, this algorithm has an exponential speedup over the fastest classical algorithm. Thus, there are attractive potential applications for the HHL algorithm, as the data size is ever growing, and solving linear systems of equations is more demanding in science and engineering. The HHL algorithm has become a subroutine in many quantum algorithms. For example, many quantum machine learning algorithms make use of the HHL algorithm. Here, we implement a simplified version of the HHL algorithm [39].

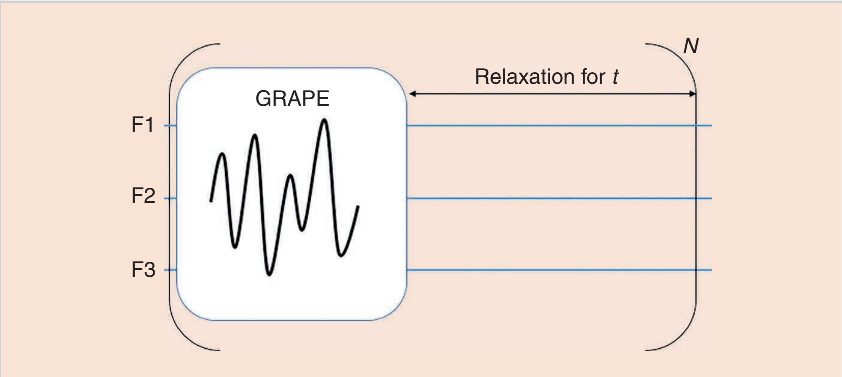


FIGURE 4 The pulse sequence for the PPS preparation. The first permutation gate is realized using a GRAPE pulse. After this is a long delay within which the natural relaxation takes effect. By properly choosing the repetition number N and the duration of the delay t, the system can be steered to the PPS $|000\rangle$ from the thermal equilibrium state. F1: first ^{19}F nuclear spin; F2: second ^{19}F nuclear spin; F3: third ^{19}F nuclear spin.

HHL ALGORITHM

The linear system of equations we are trying to solve is

$$A\vec{x} = \vec{b}, \tag{17}$$

where A is an $N \times N$ Hermitian matrix; $\vec{b}$ is a known $N \times 1$ vector; and $\vec{x}$, an unknown $N \times 1$ vector, is to be solved. To use the HHL algorithm, the first step is to express $\vec{x}$ and $\vec{b}$ in the form of quantum states, $|x\rangle$ and $|b\rangle$, which are normalized and have vectors proportional to $\vec{x}$ and $\vec{b}$, respectively. A can be considered as the matrix form of an operator, $\hat{A}$. Therefore, solving $\vec{x} = A^{-1}\vec{b}$ has been mapped to solving $|x\rangle \propto \hat{A}^{-1}|b\rangle$. $\hat{A}$ can be expressed as $\Sigma_i \lambda_i |u_i\rangle\langle u_i|$, where λ_i are eigenvalues of $\hat{A}$, and $|u_i\rangle$ are the corresponding eigenstates. The state $|b\rangle$ can be expressed using $|u_i\rangle, |b\rangle = \Sigma_i \beta_i |u_i\rangle$. Therefore, $|x\rangle$ can be written as

$$|x\rangle \propto \hat{A}^{-1}|b\rangle = \sum_i \frac{\beta_i}{\lambda_i}|u_i\rangle. \tag{18}$$

The HHL algorithm utilizes a composite system with three subsystems to derive $\Sigma_i(\beta_i/\lambda_i)|u_i\rangle$. The first subsystem contains $\log_2 N$ qubits to store $|b\rangle$ and the final result $|x\rangle$. The second subsystem is an n-qubit register used to derive the eigenvalues of $\hat{A}$. The third subsystem is an ancilla qubit that assists in deriving the reciprocal of λ_i.

There are three steps in the HHL algorithm, as shown in Figure 5. The system is initialized in the state

$$|b\rangle|0\rangle^{\otimes n}|0\rangle = \Sigma_i \beta_i |u_i\rangle|0\rangle^{\otimes n}|0\rangle. \tag{19}$$

The first step is to implement the quantum phase estimation algorithm to estimate the eigenvalues of $\hat{A}$ with an accuracy of n b. In this step, the required controlled-U operation realizes $e^{ik\hat{A}t_0/2^n}$ on the first subsystem when the n-qubit register is in the state $|k\rangle$, where $|k\rangle$ is the quantum state corresponding to the n-b binary form of k, and t_0 is usually chosen to be 2π. After this, the state of the whole system is $\Sigma_i \beta_i |u_i\rangle|\lambda_i\rangle|0\rangle$, where $|\lambda_i\rangle$ is the quantum state corresponding to the n-b binary form of λ_i.

The second step is a controlled rotation of the third subsystem. The n qubits in the register are used as the control qubits. When they are in the state $|\lambda_i\rangle$, the third subsystem is rotated by an angle of $2\sin^{-1}(C/\lambda_i)$. (Here, C is a properly chosen constant.) After this, the state of the whole system is

$$\Sigma_i \beta_i |u_i\rangle|\lambda_i\rangle\left(\sqrt{1-\left|\frac{C}{\lambda_i}\right|^2}|0\rangle + \frac{C}{\lambda_i}|1\rangle\right). \tag{20}$$

The last step is the reverse of the phase estimation. After this step, the n-qubit register is disentangled. The state of the whole system is

$$\Sigma_i \beta_i |u_i\rangle|0\rangle^{\otimes n}\left(\sqrt{1-\left|\frac{C}{\lambda_i}\right|^2}|0\rangle + \frac{C}{\lambda_i}|1\rangle\right). \tag{21}$$

Now, if the third subsystem is measured, and the result is $|1\rangle$, then the first subsystem is in the state of $\Sigma_i (C/\lambda_i)\beta_i |u_i\rangle$, which is proportional to $|x\rangle$ and, thus, is the solution. The probability of getting $|1\rangle$ of the third subsystem is $\Sigma_i |C\beta_i/\lambda_i|^2$, and this is also the probability of success.

It should be mentioned that the original article on the HHL algorithm does not give a detailed method for implementing the controlled rotation $R(\lambda^{-1})$ in the second step. The controlled unitary in the step of the quantum phase estimation can be intuitively decomposed to individual one-qubit-controlled operations using each of the register qubits as the control qubit. This is not the case for the controlled rotation $R(\lambda^{-1})$. The authors in [40] proposed a way to realize the controlled rotation $R(\lambda^{-1})$ by introducing additional ancilla qubits. Here, we do not go into the details of their method. In the three-qubit case, the $R(\lambda^{-1})$ operation can be realized in a simple way, which is described in the following section.

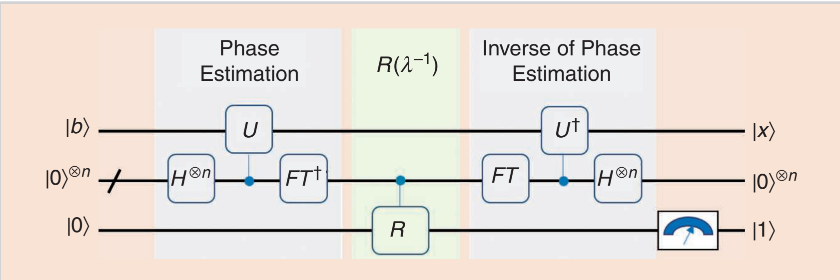


FIGURE 5 The general circuit for the HHL algorithm.

THE HHL ALGORITHM SIMPLIFIED IN THE THREE-QUBIT CASE

In a three-qubit system, we use each qubit as one of the three subsystems required by the HHL algorithm. The fact that the first subsystem has one qubit means A is a 2×2 matrix with two eigenvalues. That the second subsystem has one qubit means that, when estimating the eigenvalues of A, the accuracy is only 1 b. In the step of the quantum phase estimation, the required quantum FT can be realized using only one Hadamard gate, H.

With only one register qubit, the controlled operation in the phase estimation can be simplified as follows: when the register qubit is in $|0\rangle$, no operation is implemented as $e^{i0*At_0/2} = I$; when the register qubit is in $|1\rangle$, $e^{iAt_0/2}$ is applied to the first subsystem. After the phase estimation step, the state of the whole system is $\beta_1|u_1\rangle|0\rangle|0\rangle + \beta_2|u_2\rangle|1\rangle|0\rangle$. It can be seen that zero and one of the register qubit correspond to λ_1 and λ_2, respectively. However, each is only 1 b of λ_1 and λ_2. Therefore, in general situations, a one-qubit register for quantum phase estimation is not enough to estimate the eigenvalues of a 2×2 matrix. Some prior knowledge of λ_1 and λ_2 is required for using only one qubit in the phase estimation process.

We use the eigenvalues of two and three as an example. Two has a binary form of 10. Three has a binary form of 11. The two eigenvalues have a 1-b difference in their binary forms. Hence, a one-qubit register for phase estimation can identify them as long as we know their most significant bits are one in advance.

We also mentioned that the controlled $R(\lambda^{-1})$ rotation is difficult to realize. However, when the register has only one qubit, it becomes easy. When the register qubit is $|0\rangle$ or $|1\rangle$, apply $2\sin^{-1}(C/\lambda_1)$ or $2\sin^{-1}(C/\lambda_2)$, respectively, to the third qubit. Still using the eigenvalues two and three as an example, when the register qubit is in $|0\rangle$, we know this state corresponds to the eigenvalue 10, and the controlled $R(\lambda^{-1})$ rotation angle should be $2\sin^{-1}(C/2)$; when the register qubit is in $|1\rangle$, we know this state corresponds to the eigenvalue 11, and the controlled $R(\lambda^{-1})$ rotation angle should be $2\sin^{-1}(C/3)$.

As it concerns the choice of C, on one hand, we want the success probability $\Sigma_i |C\beta_i/\lambda_i|^2$ to be as large as possible; on the other hand, C/λ_i should be reasonable sine values. For example, in the case with $\lambda_1 = 2$ and $\lambda_2 = 3$, $C = 2$ is a good choice. In this very simple three-qubit HHL, if the initial state of the third qubit is $|1\rangle$, the controlled $R(\lambda^{-1})$ rotation step can be replaced by a $|1\rangle$-controlled $R_y(\theta)$. When the register qubit is in $|0\rangle$, no operation is needed, and, when the register qubit is in $|1\rangle$, the $R_y(\theta)$ rotation is applied to the third qubit, where $\theta = -2\cos^{-1}(\lambda_1/\lambda_2)$. From the point of view of experimental implementation, the $|1\rangle$-controlled $R_y(\theta)$ is a further simplification of the two-state-controlled $R(\lambda^{-1})$ rotation. The simplified three-qubit circuit is shown in Figure 6.

EXPERIMENTAL IMPLEMENTATION ON TRIANGULUM

Here, we realize the algorithm in the following case:

$$A = \begin{pmatrix} 2.14645 & -0.35355 \\ -0.35355 & 2.85355 \end{pmatrix}, \quad b = \begin{pmatrix} 0.70711 \\ 0.70711 \end{pmatrix}. \tag{22}$$

There are two two-qubit gates in the circuit. The $|1\rangle$-controlled $e^{iAt_0/2}$ gate can be decomposed to $U_d - CZ - Ud^{\dagger}$ upon the knowledge of the diagonalization matrix of A:

$$A = U_d^{\dagger}\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} U_d. \tag{23}$$

Here, CZ is the $|1\rangle$-controlled-σ_z gate. CZ can be combined with the two H gates and becomes the CNOT gate. The $U_d^{\dagger}$ gate in the phase estimation step and the U_d gate in the reverse of the phase estimation step can cancel each other, as shown in Figure 7. In the current case, $U_d = R^1_{-y}(\pi/4)$. Here, the superscription means the rotation is on the first qubit. The $|b\rangle$ state can be prepared using $R^1_y(\pi/2)$. The $|1\rangle$-controlled $R_y(\theta)$ operation can be realized in different ways. Here, we decompose it as

$$\mathrm{CZ} - R^3_{-x}\left(\frac{\pi}{2}\right) - R^3_{-z}\left(\frac{\pi-\theta}{2}\right) - \mathrm{CNOT} - R^3_z\left(\frac{\pi-\theta}{2}\right) - R^3_x\left(\frac{\pi}{2}\right).$$

Now, the required operations of the HHL algorithm are decomposed to basic gates that can be realized by Triangulum, and the complete gate sequence is as follows:

$$R^3_x(\pi) - R^1_y\left(\frac{\pi}{4}\right) - \mathrm{CNOT}_{12} - \mathrm{CZ}_{23} - R^3_{-x}\left(\frac{\pi}{2}\right) - R^3_{-z}\left(\frac{\pi-\theta}{2}\right) - \mathrm{CNOT}_{23} - R^3_z\left(\frac{\pi-\theta}{2}\right) - R^3_x\left(\frac{\pi}{2}\right) - \mathrm{CNOT}_{12} - R^1_y\left(\frac{\pi}{4}\right). \tag{24}$$

The first gate in this sequence prepares the third qubit in the state $|1\rangle$. It should be mentioned that the gate used to prepare $|b\rangle$ is combined with the gate U_d, and, hence, only one gate $R^1_y(\pi/4)$ is implemented in front of the first CNOT_{12} gate. The construction of this sequence using SpinQuasar is illustrated in Figure 8.

Triangulum measures the probabilities of all of the eight basis states ρ_{ii}, $i = 1\ldots 8$, as illustrated in Figure 8. Here, the subscription $i = 1\ldots 8$ corresponds to the base $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |101\rangle, |110\rangle, |111\rangle$. From the earlier analysis, we know that, when the third qubit is found in $|1\rangle$, the first qubit is in the state $|x\rangle$. Since, ideally, the final state of the second qubit is $|0\rangle$, we use ρ_{22} and ρ_{66}, which are the probabilities of $|001\rangle$ and $|101\rangle$, to infer $|x\rangle$:

$$|x\rangle \propto \begin{pmatrix} \sqrt{\rho_{22}} \\ s\sqrt{\rho_{66}} \end{pmatrix} \tag{25}$$

The result can be directly written as the square root of the probabilities because A and $\vec{b}$ are both real, and, hence, $\vec{x}$ is real as well. The term s is the relative sign between the two entries of

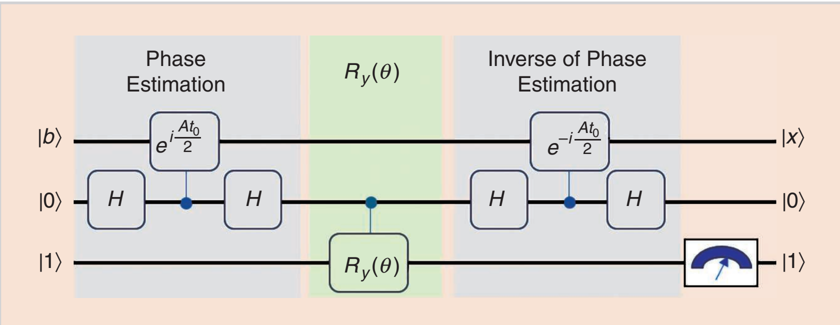


FIGURE 6 The HHL circuit in the three-qubit case.

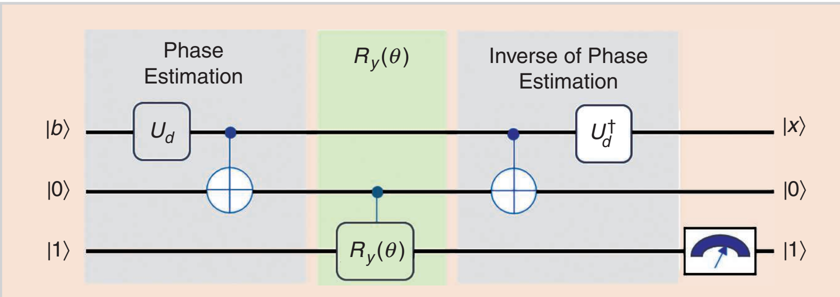


FIGURE 7 The further simplified HHL circuit upon knowledge of U_d.

FIGURE 8 The HHL sequence composed in SpinQuasar.

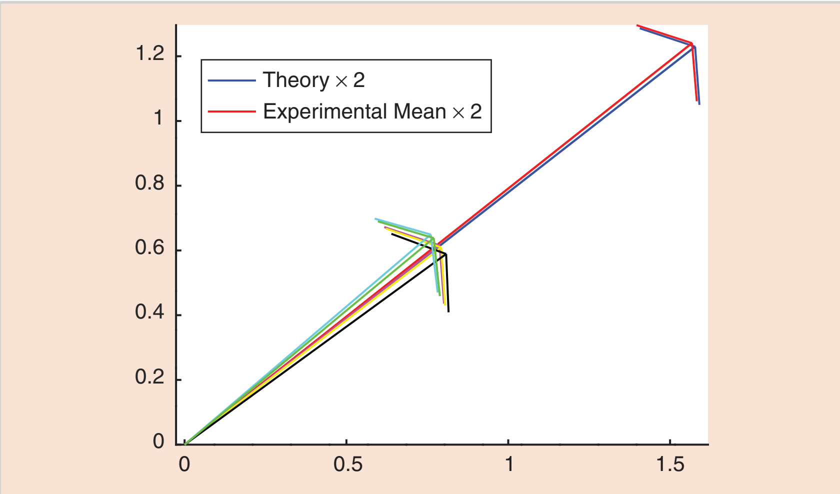


FIGURE 9 The experimental results of the five repetitions are shown as vectors. The mean vector of the five repetitions is also shown in red to compare with the theoretical vector in blue. Both the mean vector and the theoretical vector are multiplied by two for clarity.

$\vec{x}$, or $|x\rangle$, which is also the sign of the coherence term between $|001\rangle$ and $|101\rangle$, ρ_{26}, in the density matrix. This coherence term can be mapped to the measurable probabilities by a gate $R^1_{-y}(\pi/2)$. After this gate, if $\rho_{22} > \rho_{66}$, then $s=1$; otherwise, $s=-1$. Five repetitions of the experiment are done. The result vectors are shown in Figure 9. The mean vector of the results is also shown and compared with the theoretical vector; they are only ~0.4° apart, and their tangent values have a difference of only about 1.5%. The good agreement between the experimental and theoretical results implies a successful proof-of-principle demonstration of the HHL algorithm.

CONCLUSION

The successful demonstration of the HHL algorithm shows Triangulum's great potential both in quantum information education and quantum computing research. The embedded two-qubit and three-qubit quantum algorithms provide great examples for quantum computing learners. The powerful advanced pulse-control function, which provides the arbitrary-waveform-generation ability, is a great asset for research in quantum control in realistic environments for advanced users. SpinQ will continue to develop desktop quantum computing platforms with more qubits. Meanwhile, a simplified version of SpinQ Gemini, namely, Gemini Mini, has been recently realized. Gemini Mini is much more portable (at $20 \times 35 \times 26$ cm^3 and 14 kg) and affordable for most K–12 schools around the world.

ACKNOWLEDGMENTS

This work was funded by SpinQ technology. The data sets generated during and/or analyzed during the current study are available from the authors on reasonable request. We thank Prof. Ching-Ray Chang for the invitation to address quantum computers and computing in *IEEE Nanotechnology Magazine*.

ABOUT THE AUTHORS

Guanru Feng (gfeng@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Shi-Yao Hou (hshiyao@sicnu.edu.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China, and the College of Physics and Electronic Engineering and Center for Computational Sciences, Sichuan Normal University, Chengdu, 610066, China.

Hongyang Zou (hzou@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Wei Shi (wshi@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Sheng Yu (syu@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Zikai Sheng (zsheng@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Xin Rao (xrao@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Kaihong Ma (kma@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Chenxing Chen (cchen@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Bing Ren (bren@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Guoxing Miao (guo-xing.miao@uwaterloo.ca) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China, and Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada.

Jingen Xiang (jxiang@spinq.cn) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China.

Bei Zeng (zengb@ust.hk) is with Shenzhen SpinQ Technology Co., Ltd., Shenzhen, 518017, China, and the Department of Physics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, 999077, Hong Kong.

REFERENCES

[1] SpinQ. Accessed: 2018. [Online]. Available: https://www.spinq.cn

[2] "Desktop nuclear magnetic resonance quantum computer," China Patent 202021195611.3, 2020.

[3] S. Hou *et al.*, "SpinQ Gemini: A desktop quantum computing platform for education and research," *EPJ Quantum Technol.*, vol. 8, no. 1, p. 20, 2021, doi: 10.1140/epjqt/s40507-021-00109-8.

[4] D. G. Cory, A. F. Fahmy, and T. F. Havel, "Ensemble quantum computing by NMR spectroscopy," *Proc. Nat. Acad. Sci.*, vol. 94, no. 5, p. 1634, 1997, doi: 10.1073/pnas.94.5.1634.

[5] N. A. Gershenfeld and I. L. Chuang, "Bulk spin-resonance quantum computation, Computation," *Science*, vol. 275, no. 5298, p. 350, 1997, doi: 10.1126/science.275.5298.350.

[6] I. L. Chuang, N. Gershenfeld, M. G. Kubinec, and D. W. Leung, "Bulk quantum computation with nuclear magnetic resonance: Theory and experiment," *Proc. Roy. Soc. London. A, Math., Phys. Eng. Sci.*, vol. 454, no. 1969, p. 447, 1998, doi: 10.1098/rspa.1998.0170.

[7] D. G. Cory, M. D. Price, and T. F. Havel, "Nuclear magnetic resonance spectroscopy: An experimentally accessible paradigm for quantum computing," *Phys. D, Nonlinear Phenomena*, vol. 120, nos. 1–2, p. 82, 1998, doi: 10.1016/S0167-2789(98)00046-3.

[8] E. Knill and R. Laflamme, "Power of one bit of quantum information," *Phys. Rev. Lett.*, vol. 81, no. 25, p. 5672, 1998, doi: 10.1103/PhysRevLett.81.5672.

[9] D. Cory *et al.*, "NMR based quantum information processing: Achievements and prospects," *Fortschritte der Physik*, vol. 48, nos. 9–11, p. 875, 2000, doi: 10.1002/1521-3978(200009)48:9/11<875::AID-PROP875>3.0.CO;2-V.

[10] N. Linden, H. Barjat, and R. Freeman, "An implementation of the Deutsch–Jozsa algorithm on a three-qubit NMR quantum computer," *Chem. Phys. Lett.*, vol. 296, nos. 1–2, p. 61, 1998, doi: 10.1016/S0009-2614(98)01015-X.

[11] I. L. Chuang, L. M. K. Vandersypen, X. Zhou, D. W. Leung, and S. Lloyd, "Experimental realization of a quantum algorithm," *Nature*, vol. 393, no. 6681, p. 143, 1998, doi: 10.1038/30181.

[12] J. A. Jones, M. Mosca, and R. H. Hansen, "Implementation of a quantum search algorithm on a quantum computer," *Nature*, vol. 393, no. 6683, p. 344, 1998, doi: 10.1038/30687.

[13] M. A. Nielsen, E. Knill, and R. Laflamme, "Complete quantum teleportation using nuclear magnetic resonance," *Nature*, vol. 396, no. 6706, p. 52, 1998, doi: 10.1038/23891.

[14] S. Somaroo, C. H. Tseng, T. F. Havel, R. Laflamme, and D. G. Cory, "Quantum simulations on a quantum computer," *Phys. Rev. Lett.*, vol. 82, no. 26, p. 5381, 1999, doi: 10.1103/PhysRevLett.82.5381.

[15] E. Knill, R. Laflamme, R. Martinez, and C. H. Tseng, "An algorithmic benchmark for quantum information processing," *Nature*, vol. 404, no. 6776, p. 368, 2000, doi: 10.1038/35006012.

[16] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, R. Cleve, and I. L. Chuang, "Experimental realization of an order-finding algorithm with an NMR quantum computer," *Phys. Rev. Lett.*, vol. 85, no. 25, p. 5452, 2000, doi: 10.1103/PhysRevLett.85.5452.

[17] Y. S. Weinstein, M. A. Pravia, E. M. Fortunato, S. Lloyd, and D. G. Cory, "Implementation of the quantum Fourier transform," *Phys. Rev. Lett.*, vol. 86, no. 9, p. 1889, 2001, doi: 10.1103/PhysRevLett.86.1889.

[18] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, no. 6866, p. 883, 2001, doi: 10.1038/414883a.

[19] D. G. Cory *et al.*, "Experimental quantum error correction," *Phys. Rev. Lett.*, vol. 81, no. 10, p. 2152, 1998, doi: 10.1103/PhysRevLett.81.2152.

[20] E. Knill, R. Laflamme, R. Martinez, and C. Negrevergne, "Benchmarking quantum computers: The five-qubit error correcting code," *Phys. Rev. Lett.*, vol. 86, no. 25, p. 5811, 2001, doi: 10.1103/PhysRevLett.86.5811.

[21] J. A. Jones, V. Vedral, A. Ekert, and G. Castagnoli, "Geometric quantum computation using nuclear magnetic resonance," *Nature*, vol. 403, no. 6772, p. 869–871, 2000, doi: 10.1038/35002528.

[22] M. Steffen, W. van Dam, T. Hogg, G. Breyta, and I. Chuang, "Experimental implementation of an adiabatic quantum optimization algorithm," *Phys. Rev. Lett.*, vol. 90, no. 6, p. 067,903, 2003, doi: 10.1103/PhysRevLett.90.067903.

[23] H. Y. Carr and E. M. Purcell, "Effects of diffusion on free precession in nuclear magnetic resonance experiments," *Phys. Rev.*, vol. 94, no. 3, p. 630, 1954, doi: 10.1103/PhysRev.94.630.

[24] S. Meiboom and D. Gill, "Modified spin-echo method for measuring nuclear relaxation times," *Rev. Sci. Instrum.*, vol. 29, no. 8, p. 688, 1958, doi: 10.1063/1.1716296.

[25] E. M. Fortunato, M. A. Pravia, N. Boulant, G. Teklemariam, T. F. Havel, and D. G. Cory, "Design of strongly modulating pulses to implement precise effective Hamiltonians for quantum information processing," *J. Chem. Phys.*, vol. 116, no. 17, p. 7599, 2002, doi: 10.1063/1.1465412.

[26] K. R. Brown, A. W. Harrow, and I. L. Chuang, "Arbitrarily accurate composite pulse sequences," *Phys. Rev. A*, vol. 70, no. 5, p. 52,318, 2004, doi: 10.1103/PhysRevA.70.052318.

[27] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, "Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms," *J. Magnetic Resonance*, vol. 172, no. 2, p. 296, 2005, doi: 10.1016/j.jmr.2004.11.004.

[28] L. M. K. Vandersypen and I. L. Chuang, "NMR techniques for quantum control and computation," *Rev. Mod. Phys.*, vol. 76, no. 4, p. 1037, 2005, doi: 10.1103/RevModPhys.76.1037.

[29] W. G. Alway and J. A. Jones, "Arbitrary precision composite pulses for NMR quantum computing," *J. Magn. Resonance*, vol. 189, no. 1, p. 114, 2007, doi: 10.1016/j.jmr.2007.09.001.

[30] A. M. Souza, G. A. Álvarez, and D. Suter, "Robust dynamical decoupling for quantum computing and quantum memory," *Phys. Rev. Lett.*, vol. 106, no. 24, p. 240,501, 2011, doi: 10.1103/PhysRevLett.106.240501.

[31] D. Brown, B.-M. Ma, and Z. Chen, "Developments in the processing and properties of NDFEB-type permanent magnets," *J. Magn. Magn. Mater.*, vol. 248, no. 3, p. 432, 2002, doi: 10.1016/S0304-8853(02)00334-7.

[32] D. David and J. Richard, "Rapid solution of problems by quantum computation," *Proc. R. Soc. Lond. A*, vol. 439, pp. 553–558, Dec. 1992.

[33] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.* 1996, pp. 212–219, doi: 10.1145/237814.237866.

[34] G. L. Long, "Grover algorithm with zero theoretical failure rate," *Phys. Rev. A*, vol. 64, no. 2, p. 22,307, 2001, doi: 10.1103/PhysRevA.64.022307.

[35] https://cloud.spinq.cn

[36] C. A. Ryan, C. Negrevergne, M. Laforest, E. Knill, and R. Laflamme, "Liquid-state nuclear magnetic resonance as a testbed for developing quantum control methods," *Phys. Rev. A*, vol. 78, no. 1, p. 12,328, 2008, doi: 10.1103/PhysRevA.78.012328.

[37] J. Li, D. Lu, Z. Luo, R. Laflamme, X. Peng, and J. Du, "Approximation of reachable sets for coherently controlled open quantum systems: Application to quantum state engineering," *Phys. Rev. A*, vol. 94, no. 1, p. 12,312, 2016, doi: 10.1103/PhysRevA.94.012312.

[38] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, p. 150,502, 2009, doi: 10.1103/PhysRevLett.103.150502.

[39] I. M. e a KR and S. Barz, "A two-qubit photonic quantum processor and its application to solving systems of linear equations," *Sci. Rep.*, vol. 4, no. 1, p. 6115, 2014, doi: 10.1038/srep06115.

[40] Y. Cao, A. Daskin, S. Frankel, and S. Kais, "Quantum circuit design for solving linear systems of equations," *Mol. Phys.*, vol. 110, nos. 15–16, p. 1675, 2012, doi: 10.1080/00268976.2012.668289.