SpinQ Triangulum: a commercial three-qubit desktop quantum computer

Guanru Feng,1,* Shi-Yao Hou,2,† Hongyang Zou,1 Wei Shi,1 Sheng Yu,1 Zikai Sheng,1 Xin Rao,1 Kaihong Ma,1 Chenxing Chen,1 Bing Ren,1 Guoxing Miao,3,‡ Jingen Xiang,1,† and Bei Zeng4,‡

1Shenzhen SpinQ Technology Co., Ltd., Shenzhen, China
2College of Physics and Electronic Engineering & Center for Computational Sciences, Sichuan Normal University, Chengdu, China
3Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario, Canada
4Department of Physics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

(Dated: February 8, 2022)

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 room temperature NMR spectrometer, provide light-weighted, cost-effective and maintenance-free quantum computing platforms that aim to provide real-device experience for quantum computing education for K-12 and 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 QPU, smaller dimensions (61 × 33 × 56 cm³) and lighter (40 kg). Furthermore, the magnetic field is more stable and the performance of quantum control is more accurate. This paper 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 capability of undertaking complex quantum computing tasks. SpinQ will continue to develop desktop quantum computing platform with more qubits. Meanwhile, a simplified version of SpinQ Gemini, namely Gemini Mini1, has been recently realised. Gemini Mini is much more portable (20 × 35 × 26 cm³, 14 kg) and affordable for most K-12 schools around the world.

I. INTRODUCTION

SpinQ Triangulum is the second generation product of SpinQ’s commercial desktop quantum computing platform series [1–3]. Similar to the first generation product, the two-qubit SpinQ Gemini, Triangulum is based on nuclear magnetic resonance (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 × 33 × 56 cm³. Fig. 1 shows the exterior look of Triangulum and its user-friendly interface SpinQUASAR, where users can compose quantum circuits and interact with the QPU.

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 that requires 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 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 software SpinQUASAR (Fig. 1), as well as the demonstrations of 10+ famous quantum algorithms, such as Deutsch algorithm [32] and Grover algorithm [33, 34]. Thus it provides a very friendly platform for non-specialists 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.

In this paper, we introduce the system of SpinQ Triangulum. In Sec. II, we briefly introduce the hardware and software design. In Sec. III, we introduce how quantum computing is fulfilled on Triangulum. In Sec. IV, We demonstrate the implementation of Harrow-Hassidim-Lloyd (HHL) quantum algorithm on Triangulum. A discussion on future plans of next generations products will follow in Sec. V.

II. SYSTEM

The overall schematic diagram is shown in Fig. 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 arbitrary waveform generation function.

The modules that Triangulum inherits from Gemini realize

---

* gfeng@spinq.cn
† jxiang@spinq.cn
‡ zengb@ust.hk
https://www.spinq.cn/products#geminiMini-anchor
Figure 1. (a) The photo of 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 base from the experiment and the simulation are shown in the bottom half of the interface.

The NMR sample used in Triangulum is iodotrifluoroethylene (C$_2$F$_3$I). 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 excitation and signal detection for the three qubits, which seems to be a great advantage. On the other hand, 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 spins requires the pulse generator and the RF amplification/transmission system to be more powerful. In Triangulum the pulse generation module of FPGA is developed to be capable of arbitrary waveform generation, with the magnitude accuracy of 1/65536 and phase accuracy of 2$\pi$/65536. The RF system which is responsible for pulse amplification and transmission, is also improved to faithfully transmit arbitrary waveforms.

To realize a 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 Fig. 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.

III. QUANTUM COMPUTATION

A. Software interface

Same as Gemini, SpinQuasar is provided to users as the interface to Triangulum (Fig. 1). Most parts of the interface are 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 drop-down 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
Table 1. Comparison between Gemini and Triangulum.

|       | Gemini | Triangulum |
|-------|--------|------------|
| QPU   | 2 qubits | 3 qubits |
| SpinQuasar | Yes | Yes |
| Cloud service | Yes | Yes |
| API   | Yes | Yes |
| Weight | 55 kg* | 40 kg |
| Dimensions | 760×40×80* cm³ | 613×33×56 cm³ |
| RF channels | 2 | 2 |
| Arbitrary waveform generation | No | Yes |
| Field shimming | Yes | Yes |
| Temperature control | Yes | Yes |
| Field locking | No | Yes |
| Measurement time | ~3 mins | ~55 s |

*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.

Figure 2. Left: The comparison between Gemini and Triangulum. Right: The overview of the schematic diagram of Triangulum system. The master board equipped with an FPGA, provides the control logic of Triangulum. SpinQuasar communicates with 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, Preamplifier 2, which are in the orange boxes.

Figure 3. The molecule structure (left), its parameter table (right), the Fourier transform spectrum of $^{19}$F (bottom). 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 the values in the table are in unit of Hz. Each of the $^{19}$F spins has four peaks in the Fourier transform spectrum, with peak splittings determined by the J coupling constants.

### B. The spin system

The sample we use is iodotrifluoroethylene ($C_2F_3I$). The molecules are placed in the center of the 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 1 T magnetic field is $\sim$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 Fig. 3. The $T_1$ and $T_2$ for the $^{19}$F spins are about 7s and 0.2s, respectively. The J couplings between the three spins are -128 Hz, 68 Hz 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 difference between 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
\[ H_0 = 2\pi(\nu_1 I_z^1 + \nu_2 I_z^2 + \nu_3 I_z^3) \\
+ 2\pi(J_{1,2} I_z^1 I_z^2 + J_{1,3} I_z^1 I_z^3 + J_{2,3} I_z^2 I_z^3), \]  
where \( \nu_i \)'s are chemical shifts for the \( i \)th nuclei and \( J_{i,j} \)'s are \( J \) couplings between the \( i \)th and \( j \)th nuclei.

C. The gate set

The global 90 degree rotation of the three qubits can be realized using 10 us square pulses. However, global control is not enough in quantum computing. As mentioned above, on Triangulum GRAPE pulses are used to realize all the quantum gates available. The available quantum gates contain single-qubit, two-qubit and three-qubits gates. The single-qubit gates are Pauli gates \( (\sigma_x, \sigma_y, \sigma_z) \), 90 degree rotation gates and arbitrary rotation gates along \( x, y \) and \( z \) axes, Hadamard gates, \( T \) gates and inverse of \( T \) gates. The two-qubit gates are 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 CCZ gates which implement a permutation gate, are realized virtually by square pulses and delays, on Triangulum the basis \( \{ |000\rangle, |011\rangle, |100\rangle, |110\rangle, |101\rangle, |111\rangle \} \) are set for observation, which means only probability expectation values of a certain operator. Specifically, the \( R_x(\gamma) \) after the virtual \( z \) rotation is changed accordingly to \( R_{-y}(\gamma) \).

The gates in case (b), 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 \( y \).

D. The pseudo-pure state

The initial state of the three-qubit system is prepared to be a pseudo-pure state (PPS) \([4]\) using the same method as in Gemini.

A PPS has the following form,
\[ \rho_{pp} = \frac{1-\eta}{2\pi} e^{i\alpha} + \eta |\psi\rangle \langle \psi|, \]
where \( |\psi\rangle \) is a pure state. A PPS above has the same unitary dynamics and observable effects as the pure state \( |\psi\rangle \) except for the factor \( \eta \). PPS is widely used in NMR quantum computation.

Triangulum utilizes the relaxation method in Ref. [37] to prepare the three-qubit PPS starting from the thermal equilibrium state, whose state population is subject to Boltzmann distribution at room temperature. The relaxation method uses repetitions of basis permutation operations and T1 relaxation. Different from Gemini, whose permutation operation is realized by square pulses and delays, on Triangulum the basis permutation gate,
\[ U_{permute} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \]
is realized by a GRAPE pulse.

E. 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 \( |000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle \) 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 $\langle \sigma_z + i \sigma_y \rangle$ of a certain qubit. To obtain the diagonal elements of the system’s density matrix, one needs to measure $\langle \sigma_z^1 \rangle$, $\langle \sigma_z^2 \rangle$, $\langle \sigma_z^3 \rangle$, $\langle \sigma_z^1 \sigma_z^2 \rangle$, $\langle \sigma_z^1 \sigma_z^3 \rangle$, $\langle \sigma_z^2 \sigma_z^3 \rangle$, $\langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle$. The following expression illustrates how the 8 diagonal elements of the density matrix are calculated from those expectation values

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

(8)

$$
\rho_{22} = \frac{1}{8} \left[ (1 + \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right],
$$

(9)

$$
\rho_{33} = \frac{1}{8} \left[ (1 + \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right],
$$

(10)

$$
\rho_{44} = \frac{1}{8} \left[ (1 + \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right],
$$

(11)

$$
\rho_{55} = \frac{1}{8} \left[ (1 - \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right],
$$

(12)

$$
\rho_{66} = \frac{1}{8} \left[ (1 - \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right],
$$

(13)

$$
\rho_{77} = \frac{1}{8} \left[ (1 - \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right],
$$

(14)

$$
\rho_{88} = \frac{1}{8} \left[ (1 - \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle) \right].
$$

(15)

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

$$
\begin{align*}
\langle \sigma_z^1 \rangle - \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle, \\
\langle \sigma_z^1 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle, \\
\langle \sigma_z^1 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle, \\
\langle \sigma_z^1 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle.
\end{align*}
$$

(16)

By fitting the spectra and solving linear equations, $\langle \sigma_z^1 \rangle$, $\langle \sigma_z^2 \rangle$, $\langle \sigma_z^3 \rangle$, $\langle \sigma_z^1 \sigma_z^2 \rangle$, $\langle \sigma_z^1 \sigma_z^3 \rangle$, $\langle \sigma_z^2 \sigma_z^3 \rangle$, $\langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle$ can be obtained and thus the $\rho_{ii}$’s can be calculated for $i$ from 1 to 8. 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.

Figure 4. The pulse sequence for pseudo-pure state preparation. The first permutation is realized using a GRAPE pulse. After it 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 pseudo-pure state $|000\rangle$ from the thermal equilibrium state.

IV. APPLICATION: HHL ALGORITHM FOR LINEAR SYSTEMS OF EQUATIONS

In this section, we demonstrate the realization of the Harrow-Hassidim-Lloyd (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. 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 of HHL algorithm as the data size is ever growing and solving linear systems of equations is more and more demanding in science and engineering. HHL algorithm has become a subroutine in many quantum algorithms. For example, many quantum machine learning algorithms make use of HHL algorithm. Here we implement a simplified version of HHL algorithm [39].

A. HHL algorithm

The linear system of equations we are trying to solve is

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

(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 HHL algorithm, the first step is to express $\vec{x}$ and $\vec{b}$ in the form of quantum states, $|\vec{x}\rangle$ and $|\vec{b}\rangle$, 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 $|x⟩ \propto \hat{A}^{-1}|b⟩$. $\hat{A}$ can be expressed as $\Sigma_i \lambda_i |u_i⟩⟨u_i|$, where $\lambda_i$ are the eigen values of $\hat{A}$ and $|u_i⟩$ are the corresponding eigen states. $|b⟩$ can be expressed using $|u_i⟩$, $|b⟩ = \Sigma_i \beta_i |u_i⟩$. Therefore, $|x⟩$ can be written as

$$|x⟩ \propto \hat{A}^{-1}|b⟩ = \Sigma_i \beta_i |u_i⟩.$$  

(18)

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

Now if the third subsystem is measured and the result is $|1⟩$, then the first subsystem is in the state of $\Sigma_i \frac{C_{\lambda_i}}{\lambda_i} |u_i⟩$ which is proportional to $|x⟩$ and thus is the solution. The probability to get $|1⟩$ of the third subsystem is $\Sigma_i |\frac{C_{\lambda_i}}{\lambda_i}|^2$, and this is also the probability of success.

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

### B. HHL algorithm simplified in three-qubit case

In a three-qubit system, we will use each qubit as one of the three subsystems required by HHL algorithm. The fact that the first subsystem has one qubit means $A$ is a $2 \times 2$ matrix with two eigen values. That the second subsystem has one qubit means, when estimating the eigen values of $A$ the accuracy is only one bit. In the step of quantum phase estimation, the required quantum Fourier transformation (FT) can be realized using only one Hadamard gate, $H$.

![Figure 5. The general circuit for HHL algorithm.](image)

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

$$|b⟩|0⟩^n = \Sigma_i \beta_i |u_i⟩|0⟩^n = \Sigma_i \beta_i |u_i⟩|0⟩^n |0⟩.$$  

(19)

The first step is to implement the quantum phase estimation algorithm to estimate the eigen values of $\hat{A}$ with an accuracy of $n$-bits. In this step, the required controlled-$U$ operation realizes $e^{i k A t_0/2^n}$ on the first subsystem when the $n$-qubit register is in the state $|k⟩$, where $|k⟩$ is the quantum state corresponding to the $n$-bit binary form of $k$, and $t_0$ is usually chosen to be $2π$. After this, the state of the whole system is $\Sigma_i \beta_i |u_i⟩ |λ_i⟩ |0⟩$, where $|λ_i⟩$ is the quantum state corresponding to the $n$-bit 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 $|λ_i⟩$, the third subsystem is rotated by an angle $2 \sin^{-1}(C/λ_i)$ (here $C$ is a properly chosen constant). After this, the state of the whole system is

$$\Sigma_i \beta_i |u_i⟩ |λ_i⟩ (\sqrt{1 - |C/λ_i|^2} |0⟩ + \frac{C}{λ_i} |1⟩).$$  

(20)

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

$$\Sigma_i \beta_i |u_i⟩ |0⟩^n (\sqrt{1 - |C/λ_i|^2} |0⟩ + \frac{C}{λ_i} |1⟩).$$  

(21)

Now if the third subsystem is measured and the result is $|1⟩$, then the first subsystem is in the state of $\Sigma_i \frac{C}{λ_i} \beta_i |u_i⟩$ which is proportional to $|x⟩$ and thus is the solution. The probability to get $|1⟩$ of the third subsystem is $\Sigma_i |\frac{C}{λ_i} |^2$, and this is also the probability of success.

With only one register qubit, the controlled operation in phase estimation can be simplified as follows, when the register qubit is in $|0⟩$, no operation is implemented as $e^{i A t_0/2} = I$; when the register qubit is in $|1⟩$, $e^{i A t_0/2}$ is applied to the first subsystem. After the phase estimation step, the state of the whole system is $\beta_1 |u_1⟩ |0⟩ |0⟩ + \beta_2 |u_2⟩ |1⟩ |0⟩$. It can be seen that 0 and 1 of the register qubit corresponds to $λ_1$ and $λ_2$, respectively. But they each are only one bit of $λ_1$ and $λ_2$. Therefore, in general situations, a one-qubit register for quantum phase estimation is not enough to estimate the eigen values of a $2 \times 2$ 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 eigen values of 2 and 3 as an example. 2 has a binary form of 10. 3 has a binary form of 11. The two eigen values have a one-bit 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 1 in advance.

We also mentioned that the controlled $R(\lambda^{-1})$ rotation is diffi-
tion is needed and when the register qubit is in $|0\rangle$, we know this state corresponds to the eigen value 10, and the controlled $R(\lambda^{-1})$ rotation angle should be $2\sin^{-1}\frac{\lambda_{1}}{\lambda_{1}}$ when the register qubit is in $|1\rangle$, we know this state corresponds to the eigen value 11, and the controlled $R(\lambda^{-1})$ rotation angle should be $2\sin^{-1}\frac{\lambda_{1}}{\lambda_{1}}$. As concerns to the choice of $C$, on one hand we want the success probability $\Sigma_{i}|\frac{C_{ii}}{\lambda_{i}}|^{2}$ to be as large as possible, on the other hand, $C/\lambda_{i}$ 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 $\theta=-2\cos^{-1}\frac{\lambda_{2}}{\lambda_{1}}$. 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 Fig. 6.

### C. Experimental implementation on Triangulum

![Simplified HHL Circuit](image)

Figure 7. The further simplified HHL circuit upon the knowledge of $U_{d}$.

Here we realize the algorithm in the 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 $e^{iA_{0}/2}$ gate can be decomposed to $U_{d} - CZ - Ud^{\dagger}$ upon the knowledge of the diagonalization matrix of $A$,
$$A = U_{d}^{\dagger} \begin{pmatrix} A_{1} & 0 \\ 0 & A_{2} \end{pmatrix} U_{d}. \tag{23}$$

Here $CZ$ is the $|1\rangle$-controlled-$\sigma_{z}$ 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 Fig. 7. In the current case, $U_{d} = R_{y}^{\dagger}(\pi/4)$. Here, the superscription means the rotation is on the first qubit. The $|b\rangle$ state can be prepared using $R_{y}^{\dagger}(\pi/2)$. The $|1\rangle$-controlled $R_{y}(\theta)$ operation can be realized in different ways. Here, we decompose it as
$$CZ - R_{x}^{3}(\frac{\pi}{2}) - R_{z}^{3}(\frac{\pi-\theta}{2}) - \text{CNOT} - R_{x}^{3}(\frac{\pi-\theta}{2}) - R_{z}^{3}(\frac{\pi}{2}). \tag{24}$$

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
$$R_{x}^{3}(\pi) - R_{y}^{\dagger}(\frac{\pi}{4}) - \text{CNOT}_{12} - CZ_{23} - R_{x}^{3}(\frac{\pi}{2}) - R_{z}^{3}(\frac{\pi-\theta}{2}) - \text{CNOT}_{23} - R_{z}^{3}(\frac{\pi-\theta}{2}) - R_{x}^{3}(\frac{\pi}{2}) - \text{CNOT}_{12} - R_{y}^{\dagger}(\frac{\pi}{4}) \tag{25}$$

The first gate in the above 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 $R_{y}^{\dagger}(\pi/4)$ is implemented in front of the first CNOT$_{12}$ gate. The construction of this sequence using SPINQUASAR is illustrated in Fig. 8.

Triangulum measures the probabilities of all the eight basis states $\rho_{ii}, i = 1...8$, as illustrated in Fig. 8. Here the index $i = 1...8$ corresponds to the base $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, $|111\rangle$. From the analysis above, we know 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 $\rho_{22}$ and $\rho_{66}$, which are the probabilities of $|001\rangle$ and $|101\rangle$, to infer $|x\rangle$,
$$|x\rangle \propto \sqrt{\frac{\rho_{22}}{s\sqrt{\rho_{66}}}} \tag{26}$$

The result can be directly written as the square root of the probabilities is because $A$ and $b$ are both real and hence $\bar{x}$ is real as well. $s$ is the relative sign between the two entries of $\bar{x}$, or $|x\rangle$, which is also the sign of the coherence term between $|001\rangle$ and $|101\rangle$, $\rho_{26}$, in the density matrix. This coherence term can be mapped to the measurable probabilities by a gate $R_{y}^{\dagger}(\pi/4)$. After this gate, if $\rho_{26} > \rho_{66}$, then $s = 1$, otherwise $s = -1$. Five repetition of the experiments are done. The result vectors are shown in Fig. 9. The mean vector of the results is also shown and compared with the theoretical vector, which are only $\sim 0.4\sigma$ apart. And their tangent values have a difference of only about 1.5 percent. The good agreement between experimental and theoretical results implies a successful proof of principle demonstration of the HHL algorithm.

### V. 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 provides great examples for quantum computing learners. The powerful
advanced pulse control functions, which provides 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 platform with more qubits. Meanwhile, a simplified version of SpinQ Gemini, namely Gemini Mini [41], has been recently realised. Gemini Mini is much more portable (20 × 35 × 26 cm³, 14 kg) and affordable for most K-12 schools around the world.

[1] https://www.spinq.cn ()
[2] Desktop nuclear magnetic resonance quantum computer (China Patent ZL 2020 2 1195611.3, 2020).
[3] S. Hou, G. Feng, Z. Wu, and et al., Spinq gemini: a desktop quantum computing platform for education and research, EPJ Quantum Technol. 8, 20 (2021).
[4] D. G. Cory, A. F. Fahmy, and T. F. Havel, Ensemble quantum computing by nmr spectroscopy, Proceedings of the National Academy of Sciences 94, 1634 (1997).
[5] N. A. Gershenfeld and I. L. Chuang, Bulk spin-resonance quantum computation, Computation.” Science 275, 350 (1997).
[6] I. L. Chuang, N. Gershenfeld, M. G. Kubinec, and D. W. Leung, Bulk quantum computation with nuclear magnetic resonance: theory and experiment, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454, 447 (1998).
[7] D. G. Cory, M. D. Price, and T. F. Havel, Nuclear magnetic resonance spectroscopy: An experimentally accessible paradigm for quantum computing, Physica D: Nonlinear Phenomena 120, 82 (1998), proceedings of the Fourth Workshop on Physics and Consumption.
[8] E. Knill and R. Laflamme, Power of one bit of quantum information, Phys. Rev. Lett. 81, 5672 (1998).
[9] D. Cory, R. Laflamme, E. Knill, L. Viola, T. Havel, N. Boulant, G. Boutris, E. Fortunato, S. Lloyd, R. Martinez, C. Ne- grevğerne, M. Pravia, Y. Sharf, G. Teklemariam, Y. Weinstein, and W. Zurek, Nmr based quantum information processing: Achievements and prospects, Fortschritte der Physik 48, 875 (2000).
[10] N. Linden, H. Barjat, and R. Freeman, An implementation of the deutsch–jozsa algorithm on a three-qubit nmr quantum computer, Chemical Physics Letters 296, 61 (1998).
[11] I. L. Chuang, L. M. K. Vandersypen, X. Zhou, D. W. Leung, and S. Lloyd, Experimental realization of a quantum algorithm, Nature 393, 143 (1998).
[12] J. A. Jones, M. Mosca, and R. H. Hansen, Implementation of a quantum search algorithm on a quantum computer, Nature 393, 344 (1998).
[13] M. A. Nielsen, E. Knill, and R. Laflamme, Complete quantum teleportation using nuclear magnetic resonance, Nature 396, 52 (1998).
[14] S. Somaroo, C. H. Tseng, T. F. Havel, R. Laflamme, and D. G. Cory, Quantum simulations on a quantum computer, Phys. Rev. Lett. 82, 5381 (1999).
[15] E. Knill, R. Laflamme, R. Martinez, and C. H. Tseng, An algorithmic benchmark for quantum information processing, Nature 404, 368 (2000).
[16] L. M. K. Vandersypen, M. Steffen, G. Breyma, C. S. Yannoni, R. Cleve, and I. L. Chuang, Experimental realization of an order-finding algorithm with an nmr quantum computer, Phys.
Figure 9. The experimental results of the five repetitions are shown as the vectors. The mean vector of the five repetitions are also shown in red to compare with the theoretical vector in blue. Both the mean vector and the theoretical vector are multiplied by 2 for clarity.

Rev. Lett. 85, 5452 (2000).
[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. 86, 1889 (2001).
[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 414, 883 (2001).
[19] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo, Experimental quantum error correction, Phys. Rev. Lett. 81, 2152 (1998).
[20] E. Knill, R. Laflamme, R. Martinez, and C. Negrevergne, Benchmarking quantum computers: The five-qubit error correcting code, Phys. Rev. Lett. 86, 5811 (2001).
[21] J. A. Jones, V. Vedral, A. Ekert, and G. Castagnoli, Geometric quantum computation using nuclear magnetic resonance, Nature 403, 869–871 (2000).
[22] M. Steffen, W. van Dam, T. Hogg, G. Breyta, and I. Chuang, Experimental implementation of an adiabatic quantum optimization algorithm, Phys. Rev. Lett. 90, 067903 (2003).
[23] H. Y. Carr and E. M. Purcell, Effects of diffusion on free precession in nuclear magnetic resonance experiments, Phys. Rev. 94, 630 (1954).
[24] S. Meiboom and D. Gill, Modified spin-echo method for measuring nuclear relaxation times, Review of Scientific Instruments 29, 688 (1958).
[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, The Journal of Chemical Physics 116, 7599 (2002).
[26] K. R. Brown, A. W. Harrow, and I. L. Chuang, Arbitrarily accurate composite pulse sequences, Phys. Rev. A 70, 052318 (2004).
[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, Journal of Magnetic Resonance 172, 296 (2005).
[28] L. M. K. Vandersypen and I. L. Chuang, Nmr techniques for quantum control and computation, Rev. Mod. Phys. 76, 1037 (2005).
[29] W. G. Alway and J. A. Jones, Arbitrary precision composite pulses for nmr quantum computing, Journal of Magnetic Resonance 189, 114 (2007).
[30] A. M. Souza, G. A. Álvarez, and D. Suter, Robust dynamical decoupling for quantum computing and quantum memory, Phys. Rev. Lett. 106, 240501 (2011).
[31] D. Brown, B.-M. Ma, and Z. Chen, Developments in the processing and properties of ndfeb-type permanent magnets, Journal of Magnetism and Magnetic Materials 248, 432 (2002).
[32] D. David and J. Richard, Rapid solution of problems by quantum computation, Proc. R. Soc. Lond. A 439, 553–558 (1992).
[33] L. K. Grover, A fast quantum mechanical algorithm for database search, in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (1996) pp. 212–219.
[34] G. L. Long, Grover algorithm with zero theoretical failure rate, Phys. Rev. A 64, 022307 (2001).
[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 78, 012328 (2008).
[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 94, 012312 (2016).
[38] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. 103, 150502 (2009).
[39] S. Barz, I. Kassal, M. Ringbauer, and et al, A two-qubit photonic quantum processor and its application to solving systems of linear equations, Sci Rep 4, 6115 (2014).
[40] Y. Cao, A. Daskin, S. Frankel, and S. Kais, Quantum circuit design for solving linear systems of equations, Molecular Physics 110, 1675 (2012).
[41] https://www.spinq.cn/products#geminiMini-anchor.