Algorithms in Superconducting Circuits

Axel Dahlberg, Marco Roth

April 24, 2015

Motivation

1. Implementing Grover's algorithm.

Motivation

1. Implementing Grover's algorithm.
2. Superconducting circuits.

Motivation

1. Implementing Grover's algorithm.
2. Superconducting circuits.
3. Control over parameters.

Outline

Grover's Algorithm

1. A quantum algorithm for finding an element in an unstructured set of $N = 2^n$ elements.

Grover's Algorithm

1. A quantum algorithm for finding an element in an unstructured set of $N = 2^n$ elements.
2. **Assumption**: Given a function $f(x) : \{0, 1\}^n \to \{0, 1\}$ such that

$$f(x) = \begin{cases} 1 & x = x_k \\ 0 & else \end{cases}, \tag{1}$$

for some k.

Grover's Algorithm

1. A quantum algorithm for finding an element in an unstructured set of $N = 2^n$ elements.
2. **Assumption**: Given a function $f(x) : \{0,1\}^n \to \{0,1\}$ such that

$$f(x) = \begin{cases} 1 & x = x_k \\ 0 & else \end{cases}, \tag{1}$$

   for some k.
3. **Task**: Find x_k with the least amount of calls to the function f.

Grover's Algorithm

1. A quantum algorithm for finding an element in an unstructured set of $N = 2^n$ elements.
2. **Assumption**: Given a function $f(x) : \{0,1\}^n \to \{0,1\}$ such that

$$f(x) = \begin{cases} 1 & x = x_k \\ 0 & else \end{cases} , \tag{1}$$

   for some k.
3. **Task**: Find x_k with the least amount of calls to the function f.
4. To evaluate the function f a gate called the oracle O_f is implemented. The gate does the following

$$|x\rangle \to (-1)^{f(x)} |x\rangle . \tag{2}$$

Grover's Algorithm

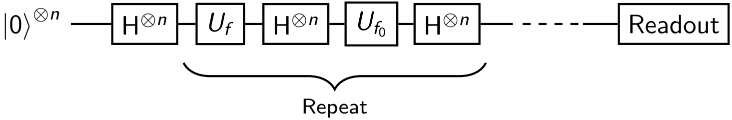
1. In classical computation the task can only be completed with $\mathcal{O}(N)$ calls to the oracle.

Grover's Algorithm

1. In classical computation the task can only be completed with $\mathcal{O}(N)$ calls to the oracle.
2. Grover's algorithm does this with $\mathcal{O}(\sqrt{N})$ calls.

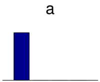
Grover's Algorithm

1. In classical computation the task can only be completed with $\mathcal{O}(N)$ calls to the oracle.
2. Grover's algorithm does this with $\mathcal{O}(\sqrt{N})$ calls.



The function f_0 outputs a 1 if the input is $0^{\otimes n}$ and otherwise 0.

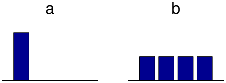
Grover's Algorithm



$$|\varphi_a\rangle = |0,0\rangle$$

One iteration of Grover's algorithm on two qubits.

Grover's Algorithm



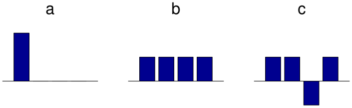
$$|\varphi_{\mathsf{a}}\rangle = |0,0\rangle$$

$$\downarrow \quad \mathsf{H}^{\otimes 2}$$

$$|\varphi_{\mathsf{b}}\rangle = (|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

One iteration of Grover's algorithm on two qubits.

Grover's Algorithm



$$|\varphi_{\mathrm{a}}\rangle = |0,0\rangle$$

$$\downarrow \quad \mathrm{H}^{\otimes 2}$$

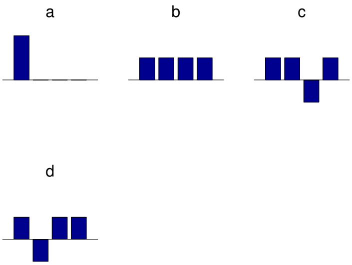
$$|\varphi_{\mathrm{b}}\rangle = (|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

$$\downarrow \quad U_f$$

$$|\varphi_{\mathrm{c}}\rangle = (|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle)/2$$

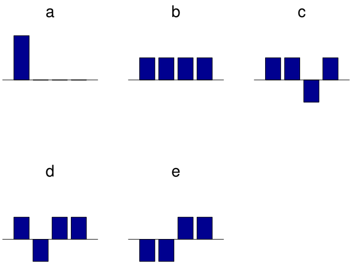
One iteration of Grover's algorithm on two qubits.

Grover's Algorithm



$$|\varphi_a\rangle = |0,0\rangle$$
$$\Big\downarrow \quad H^{\otimes 2}$$
$$|\varphi_b\rangle = (|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$
$$\Big\downarrow \quad U_f$$
$$|\varphi_c\rangle = (|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle)/2$$
$$\Big\downarrow \quad H^{\otimes 2}$$
$$|\varphi_d\rangle = (|0,0\rangle - |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

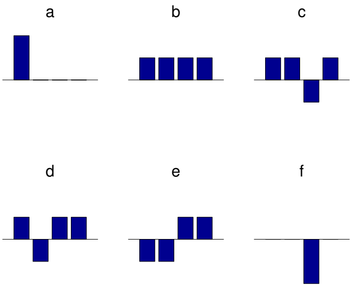
One iteration of Grover's algorithm on two qubits.



$$|\varphi_{\mathrm{a}}\rangle = |0,0\rangle$$
$$\downarrow \quad \mathrm{H}^{\otimes 2}$$
$$|\varphi_{\mathrm{b}}\rangle = (|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$
$$\downarrow \quad U_f$$
$$|\varphi_{\mathrm{c}}\rangle = (|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle)/2$$
$$\downarrow \quad \mathrm{H}^{\otimes 2}$$
$$|\varphi_{\mathrm{d}}\rangle = (|0,0\rangle - |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$
$$\downarrow \quad U_{f_0}$$
$$|\varphi_{\mathrm{e}}\rangle = (-|0,0\rangle - |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

One iteration of Grover's algorithm on two qubits.

Grover's Algorithm



$$|\varphi_{\mathsf{a}}\rangle = |0,0\rangle$$

$$\downarrow \quad \mathsf{H}^{\otimes 2}$$

$$|\varphi_{\mathsf{b}}\rangle = (|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

$$\downarrow \quad U_f$$

$$|\varphi_{\mathsf{c}}\rangle = (|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle)/2$$

$$\downarrow \quad \mathsf{H}^{\otimes 2}$$

$$|\varphi_{\mathsf{d}}\rangle = (|0,0\rangle - |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

$$\downarrow \quad U_{f_0}$$

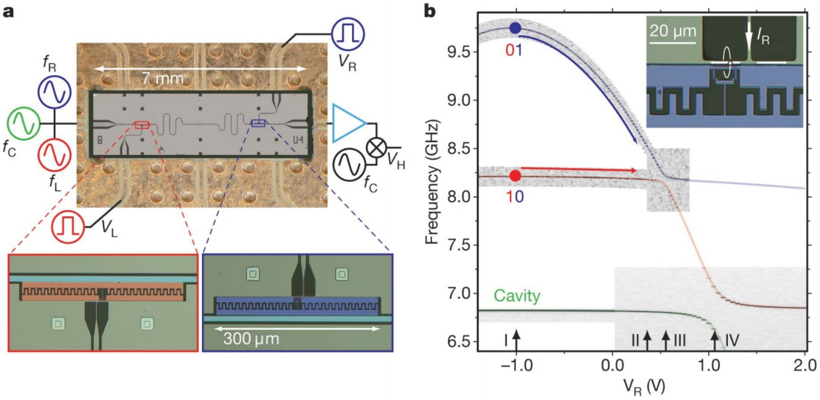
$$|\varphi_{\mathsf{e}}\rangle = (-|0,0\rangle - |0,1\rangle + |1,0\rangle + |1,1\rangle)/2$$

$$\downarrow \quad \mathsf{H}^{\otimes 2}$$

$$|\varphi_{\mathsf{f}}\rangle = -|1,0\rangle$$

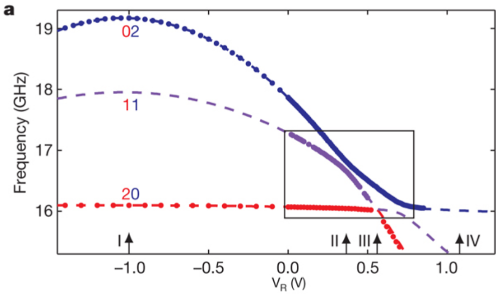
One iteration of Grover's algorithm on two qubits.

Superconducting circuit with two qubits used for realising Grover's algorithm.



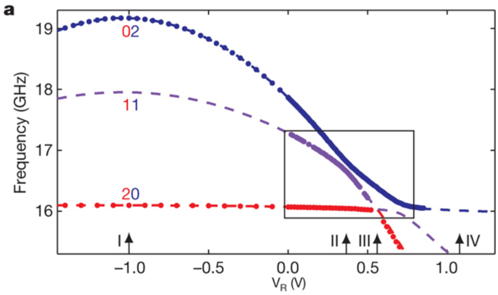
A four-port device with a coplanar waveguide cavity bus coupling two transmon qubits (Fig. taken from [1]).

Implementing the C-Phase gate



Flux dependence of transition frequencies (Fig. taken from [1]).

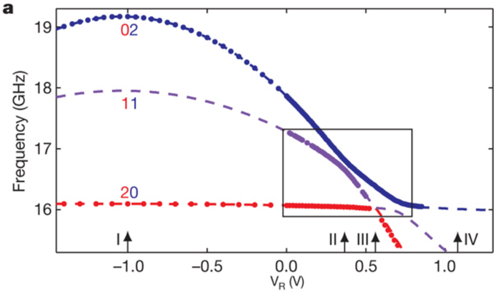
Implementing the C-Phase gate



Flux dependence of transition frequencies (Fig. taken from [1]).

1. Cavity-qubit interactions produce a frequency shift between the $|1,1\rangle$ and the $|0,2\rangle$ state

Implementing the C-Phase gate



Flux dependence of transition frequencies (Fig. taken from [1]).

1. Cavity-qubit interactions produce a frequency shift between the $|1,1\rangle$ and the $|0,2\rangle$ state
2. This shift can be exploited to create a C-Phase gate:

Implementing the C-Phase gate

1. The interaction between the $|1,1\rangle$ state and the non-computational $|0,2\rangle$ state introduces a phase difference:

Implementing the C-Phase gate

1. The interaction between the $|1,1\rangle$ state and the non-computational $|0,2\rangle$ state introduces a phase difference:

Implementing the C-Phase gate

1. The interaction between the $|1,1\rangle$ state and the non-computational $|0,2\rangle$ state introduces a phase difference:

$$\frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle + 0\,|0,2\rangle)$$

Implementing the C-Phase gate

1. The interaction between the $|1,1\rangle$ state and the non-computational $|0,2\rangle$ state introduces a phase difference:

$$\frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle + 0\,|0,2\rangle)$$
$$\xrightarrow{\text{flux pulse}} \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + e^{-i\pi}\,|1,1\rangle + e^{+i\pi}0\,|0,2\rangle)$$

Implementing the C-Phase gate

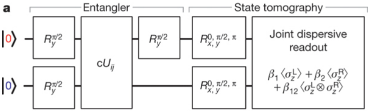
1. The interaction between the $|1,1\rangle$ state and the non-computational $|0,2\rangle$ state introduces a phase difference:

$$\frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle + 0\,|0,2\rangle)$$
$$\xrightarrow{\text{flux pulse}} \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + e^{-i\pi}\,|1,1\rangle + e^{+i\pi}0\,|0,2\rangle)$$

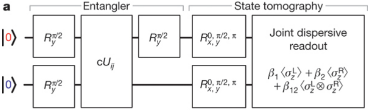
2. Which effectively produces the gate:

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (3)$$

Creating entanglement

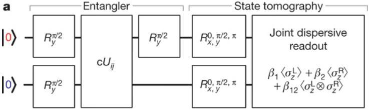


Creating entanglement



$$|\varphi_a\rangle = |0,0\rangle$$

Creating entanglement

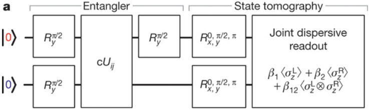


$$|\varphi_a\rangle = |0,0\rangle$$

$$\downarrow \quad R_y^{\pi/2} \otimes R_y^{\pi/2}$$

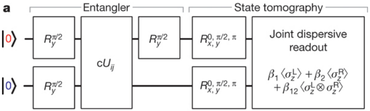
$$|\varphi_b\rangle = \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)$$

Creating entanglement



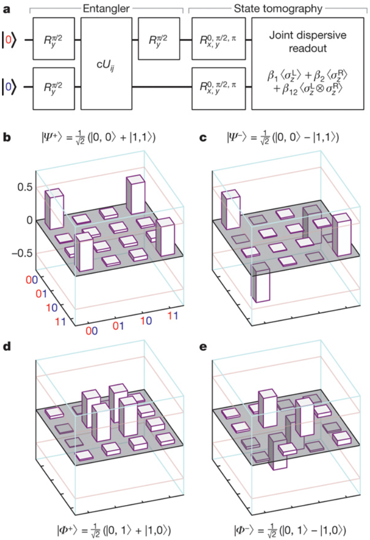
$$|\varphi_a\rangle = |0,0\rangle$$
$$\downarrow \quad R_y^{\pi/2} \otimes R_y^{\pi/2}$$
$$|\varphi_b\rangle = \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)$$
$$\downarrow \quad U_{10}$$
$$|\varphi_c\rangle = \frac{1}{2}(|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle)$$

Creating entanglement



$$
\begin{aligned}
|\varphi_{\mathsf{a}}\rangle &= |0,0\rangle \\
&\downarrow \quad \mathrm{R}_y^{\pi/2} \otimes \mathrm{R}_y^{\pi/2} \\
|\varphi_{\mathsf{b}}\rangle &= \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle) \\
&\downarrow \quad U_{10} \\
|\varphi_{\mathsf{c}}\rangle &= \frac{1}{2}(|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle) \\
&\downarrow \quad \mathrm{R}_y^{\pi/2} \\
|\varphi_{\mathsf{d}}\rangle &= \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle) = \left|\Psi^+\right\rangle
\end{aligned}
$$

Creating entanglement



$$|\varphi_a\rangle = |0,0\rangle$$

$$\downarrow \quad R_y^{\pi/2}\otimes R_y^{\pi/2}$$

$$|\varphi_b\rangle = \frac{1}{2}(|0,0\rangle+|0,1\rangle+|1,0\rangle+|1,1\rangle)$$

$$\downarrow \quad U_{10}$$

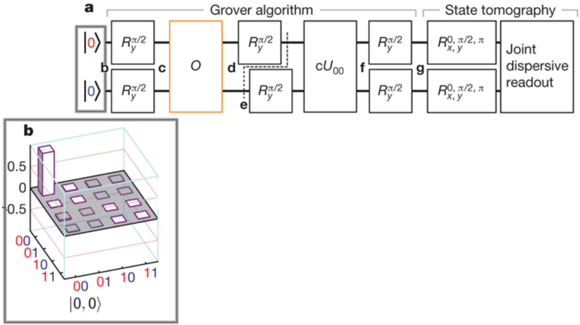
$$|\varphi_c\rangle = \frac{1}{2}(|0,0\rangle+|0,1\rangle-|1,0\rangle+|1,1\rangle)$$

$$\downarrow \quad R_y^{\pi/2}$$

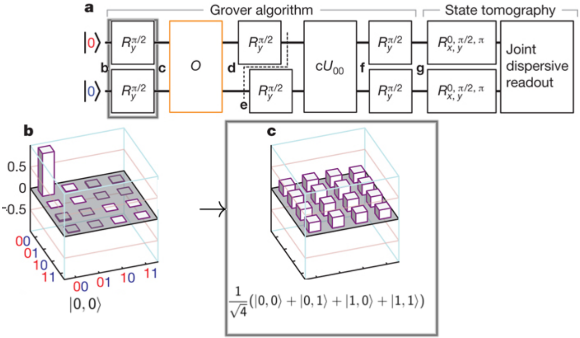
$$|\varphi_d\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle+|1,1\rangle) = |\Psi^+\rangle$$

The C-Phase gate can be used to create entanglement (Fig. taken from [1]).

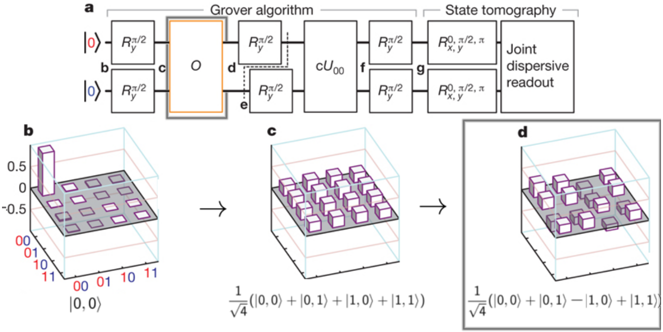
Grover's algorithm

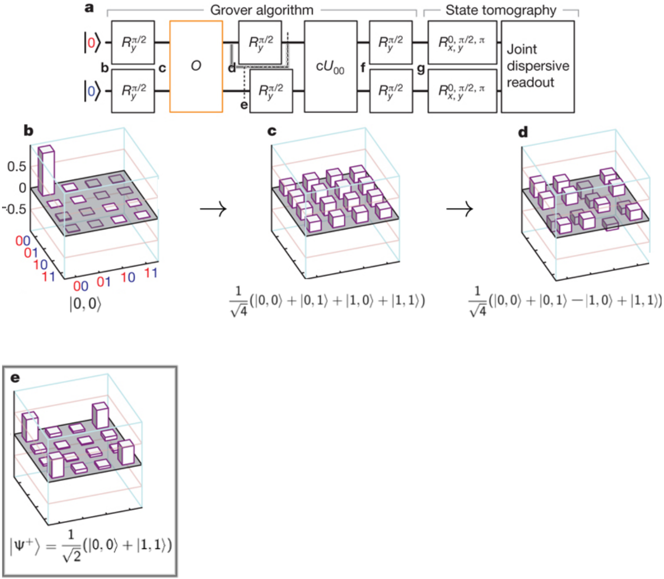


Grover's algorithm

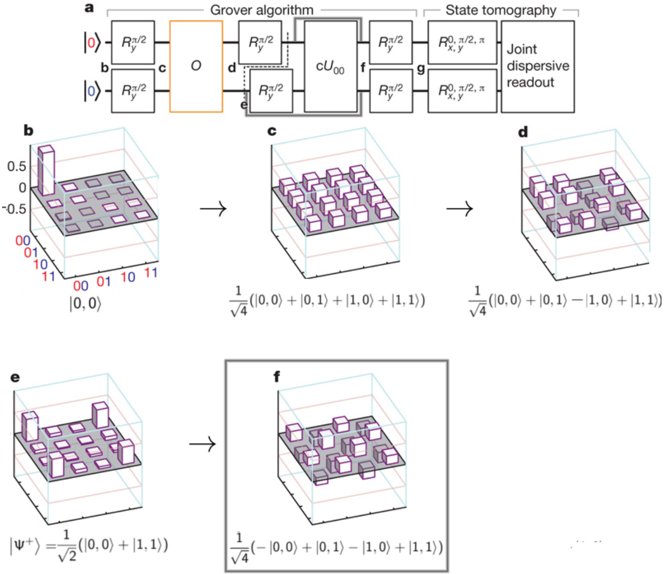


Grover's algorithm

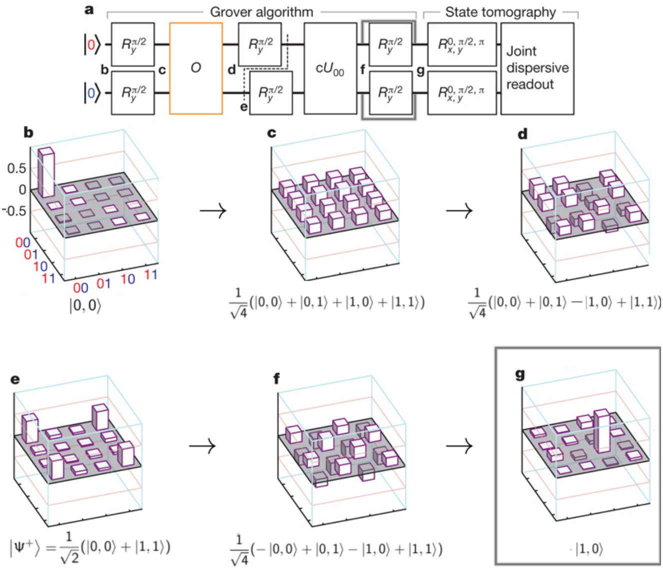



a
Grover algorithm
State tomography
|0⟩
|0⟩
$R_y^{\pi/2}$
O
cU00
$R_{x,y}^{0,\pi/2,\pi}$
Joint dispersive readout
b
c
d
e
f
g
0.5
0
-0.5
00
01
10
11
$|0,0\rangle$
$\frac{1}{\sqrt{4}}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)$
$\frac{1}{\sqrt{4}}(|0,0\rangle + |0,1\rangle - |1,0\rangle + |1,1\rangle)$
$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)$

Grover's algorithm



Grover's algorithm



Fidelity of Results

Table 1 | Summary of algorithmic performance

Element		Grover search oracle*			
		f_{00}	f_{01}	f_{10}	f_{11}
$\langle 0,0 \vert \rho \vert 0,0 \rangle$	Ideal	1	0	0	0
	Measured	0.81(1)	0.08(1)	0.07(2)	0.065(7)
$\langle 0,1 \vert \rho \vert 0,1 \rangle$	Ideal	0	1	0	0
	Measured	0.066(7)	0.802(9)	0.05(1)	0.054(8)
$\langle 1,0 \vert \rho \vert 1,0 \rangle$	Ideal	0	0	1	0
	Measured	0.08(1)	0.05(1)	0.82(2)	0.07(1)
$\langle 1,1 \vert \rho \vert 1,1 \rangle$	Ideal	0	0	0	1
	Measured	0.05(2)	0.07(1)	0.06(1)	0.81(1)

Fidelity $F(\rho, \psi) = \langle\psi| \rho |\psi\rangle$ of final states of Grover's algorithm (Figure taken from [1]).

Conclusion

1. The on-demand creation and detection of entangled states is possible.

Conclusion

1. The on-demand creation and detection of entangled states is possible.
2. Basic algorithms have been implemented in superconducting circuit systems.

Conclusion

1. The on-demand creation and detection of entangled states is possible.
2. Basic algorithms have been implemented in superconducting circuit systems.
3. The present architecture can easily be expanded to several qubits.

Sources

1 L. DiCarlo, J. M. Chow, J. M. Gambetta, Lev S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin & R. J. Schoelkopf
Demonstration of two-qubit algorithms with a superconducting quantum processor
Nature 460, 7252 (2009)

2 Clarke, J. & Wilhelm, F.K.
Superconducting quantum bits
Nature 453, 1031 (2008)

3 Schoelkopf, R.J. & Girvin, S.M.
Wiring up quantum systems
Nature 451, 664 (2008)

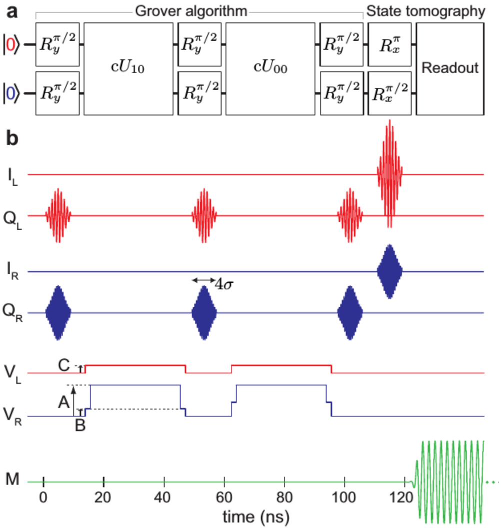
4 Devoret, M.H. & Martinis, J.M. *Implementing Qubits with Superconducting Integrated Circuits*
Quant. Inf. Proc, 3 163 (2004)

Creating Phase Gates

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\Phi_{01}} & 0 & 0 \\ 0 & 0 & e^{i\Phi_{10}} & 0 \\ 0 & 0 & 0 & e^{i\Phi_{11}} \end{pmatrix} \tag{4}$$

- Φ_{01} is adjusted by tuning the rising or falling edge of the pulse
- Φ_{10} is adjusted by varying the amplitude of a simultaneous V_L pulse

Creating Phase Gates



Creating U_{01}