Quantum Gates and Circuits

Parth Deshmukh, Andrey Vlasov

Outline

Important notes

- This is part 2 of 2 in a series on quantum error correction. Hopefully you enjoyed the last one!
- Some advice: when learning quantum mechanics, treat it like a game with rules to play. (Unless you're a physicist.)

Quick recap

- We represent the states of qubits as vectors in $\mathbb{C}^2$ using bra-ket notation
- We can do everything to them that we can do to vectors
- Most importantly, we can measure them: $|\langle 0|\psi\rangle|^2$ tells us the probability we get 0, and $|\langle 1|\psi\rangle|^2$ the probability we get 1

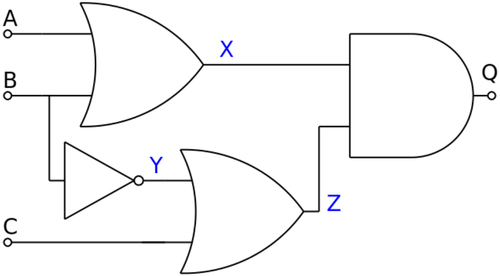
Section 1

Quantum circuits

What are circuits?

- Circuits in classical computing are what EEs and CEs study
- They have wires that carry individual bits and gates that take in those wires and modify their bits
- The language of circuits is Boolean logic

A classical circuit



A	B	C	X	Y	Z	Q
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	1	0	0	0
0	1	1	1	0	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	0	0	0
1	1	1	1	0	1	1

Figure: X and Z are OR, Y is NOT, Z is NAND (AND then NOT)

Classical to quantum circuits

- Quantum circuits operate exactly the same way; they have wires carrying qubits and quantum gates
- There's one important rule to contend with: *we can reverse time*
- Not actually, but the laws governing quantum mechanics are the same whether time moves forwards or backwards
- So we need the same amount of wires going out and in, and our gates need to be reversible - in fact, they need to be unitary

A quantum circuit

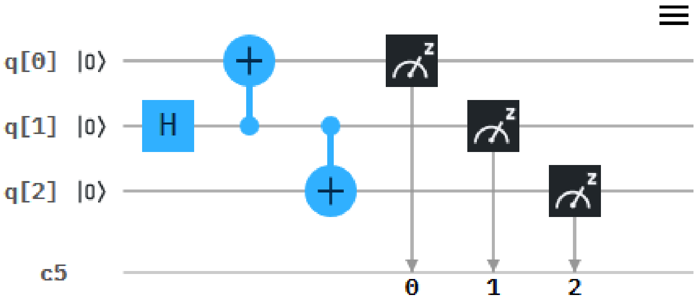


Figure: We have three wires going in and out, called the quantum register, and a classical register to collect the outcome of measurements.

Section 2

Quantum Gates

Basic Definition

- A quantum gate *acts* on a qubit
- In other words, a quantum gate changes the state of a qubit
- There are a few properties we want quantum gates to have
- The reason for these properties: visually, single-qubit quantum gates are rotations on the Bloch sphere

Quantum Gate Properties

- Recall that a quantum state of a qubit is given by a superposition of our chosen ONB
- In quantum physics, we want the total probability to always be equal to 1
- Therefore, when the state of a qubit changes, we want the change to be *ditsributive* across superpositions and *normalized*
- Thus every quantum gate U must be a normalized linear map

What is a linear map?

- Roughly, a function that preserves operations and linear combinations
- Recall that a quantum state is a linear combination of our two basis states
- $U(\alpha|0\rangle + \beta|1\rangle) = \alpha U(|0\rangle) + \beta U(|1\rangle)$
- Normalization: $|\alpha|^2 + |\beta|^2 = 1$

Why is it called U?

- Every valid quantum gate can be represented as a Unitary matrix, and every unitary matrix is a quantum gate
- In bra-ket notation, $\langle\psi|\psi\rangle = \psi^{*\intercal}\psi = 1$ by Born's Rule
- A unitary matrix is defined as $U^{*\intercal}U = U^{\dagger}UI$
- $\langle U\psi| = (|U\psi\rangle)^{\dagger} = (U|\psi\rangle)^{\dagger} = |\psi\rangle^{\dagger}U^{\dagger} = \langle\psi|U^{\dagger}$
- $\langle U\psi|U\psi\rangle = \langle\psi|U^{\dagger}U|\psi\rangle = \langle\psi|\psi\rangle = 1$

Subsection 1

Specific Gates

The Hadamard Gate

- $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$
- Like any quantum gate, the Hadamard gate is reversible
- $H|+\rangle = |0\rangle$, $H|-\rangle = |1\rangle$
- Corresponds to a 180° rotation about the $x + z$-axis on the Bloch sphere
- So H^2 is a 360° rotation, which is just I

Understanding the Hadamard gate

- The actual Hadamard gate is $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- If we let the Hadamard gate act on a qubit $|0\rangle$ and then measure it, what do we get?
- The changed qubit is written as $H|0\rangle$, and if we measure it, our probabilities are written $|\langle 0|H|0\rangle|^2$ and $|\langle 1|H|0\rangle|^2$
- If we work it out, we get exactly $1/2$ for both! Remember that $H|0\rangle = |+\rangle$, so we get $|\langle 0|+\rangle|^2 = (\frac{1}{\sqrt{2}})^2$ (and same for $|\langle 1|+\rangle|^2$)
- This also works for $|1\rangle$, so the point of the Hadamard gate is to put a qubit exactly between $|0\rangle$ and $|1\rangle$ on the Bloch sphere

The Pauli-X gate

- Another useful single-qubit gate
- Represented as $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- The Pauli-X gate is simply a 180 degree rotation about the X axis of the Bloch sphere
- Which means it's a bit flip: $X|0\rangle = |1\rangle$, and $X|1\rangle = |0\rangle$

Multiple-qubit gates

- Gates can operate on more than one qubit: for example, CNOT operates on two gates, and CCNOT operates on three
- These are represented as 4x4 and 8x8 matrices, respectively, which are unitaries in the wider spaces $\mathbb{C}^4$ and $\mathbb{C}^8$
- Why are we doing this? Because the input qubits to multi-qubit quantum gates need to be combined with the tensor product, since they could become entangled
- Example: input qubits $|0\rangle$ and $|1\rangle$, so the input to the CNOT gate would be the tensor product $|01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

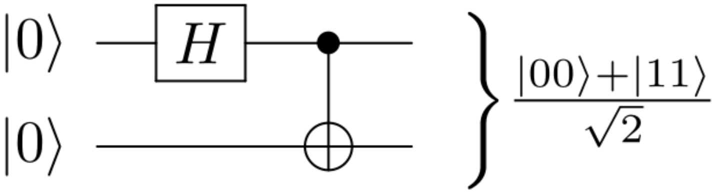
The CNOT gate

- CNOT = controlled-NOT
- Flips the second qubit only if the first qubit is $|1\rangle$, does nothing if it's $|0\rangle$
- CNOT gate is defined then as $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- First qubit is the control qubit, second qubit is the target qubit
- If we have different qubits than $|0\rangle$ and $|1\rangle$, the behavior is analogous but more complex

CNOT entanglement

- CNOT gates are super useful to entangle bits together
- Since they're the simplest gate that acts on two qubits, they're the simplest way to tie two qubits together - which is where the power of quantum algorithms lie
- The simplest way of making a maximally entangled state uses a Hadamard and CNOT to make what's called the Bell state

Creating the Bell state



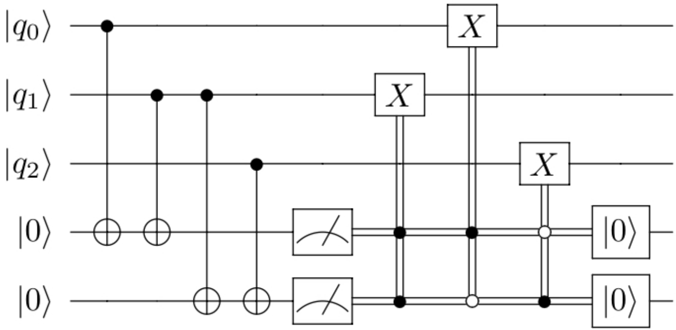
Section 3

Putting it all together

Our goal

- Recall we want to find a way to do quantum error correction
- If a qubit is at $|0\rangle$ or $|1\rangle$ and drifts from it, that's a partial or complete bit flip
- We'll use backup qubits, CNOT and Pauli-X gates, and measurement to indirectly force the qubit back onto $|0\rangle$ or $|1\rangle$ and then correct the error

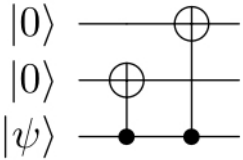
The circuit



$|q_0\rangle$
$|q_1\rangle$
$|q_2\rangle$
$|0\rangle$
$|0\rangle$
X
X
X
$|0\rangle$
$|0\rangle$

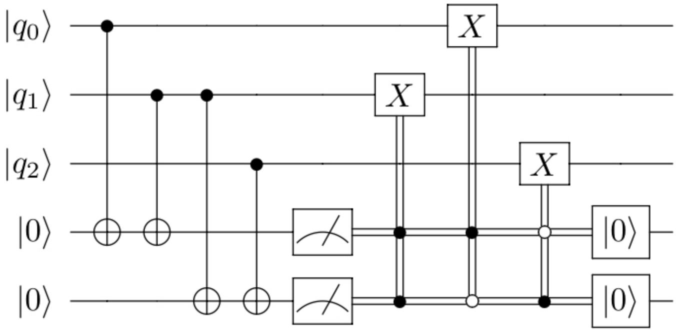
Step 1: encoding a qubit

- We'll use three physical qubits to encode each logical qubit
- If our starting qubit is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we want to turn it into $\alpha|000\rangle + \beta|111\rangle$
- This isn't shown in the figure, but we can do this with two CNOT gates:



- Note we're not cloning our qubit, so we respect no-cloning theorem

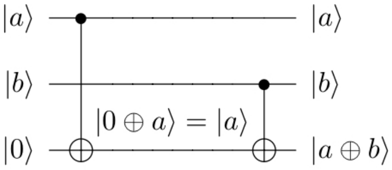
The circuit



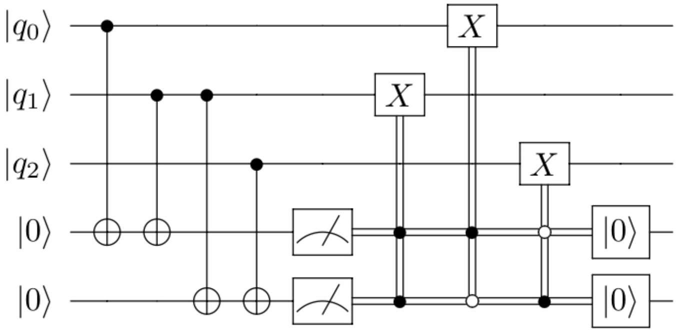
$|q_0\rangle$
$|q_1\rangle$
$|q_2\rangle$
$|0\rangle$
$|0\rangle$
X
X
X
$|0\rangle$
$|0\rangle$

Step 2: measuring parities

- $|q_0\rangle, |q_1\rangle,$ and $|q_2\rangle$ are the three qubits from our encoded qubit $|\psi\rangle$
- The parity of two bits is 1 if they differ, and 0 if they don't
- The parity of two bits is their XOR $a \oplus b$. We implement this with CNOT, since $CNOT|a\rangle|b\rangle = |a\rangle|a \oplus b\rangle$, and put it in an backup qubit



The circuit

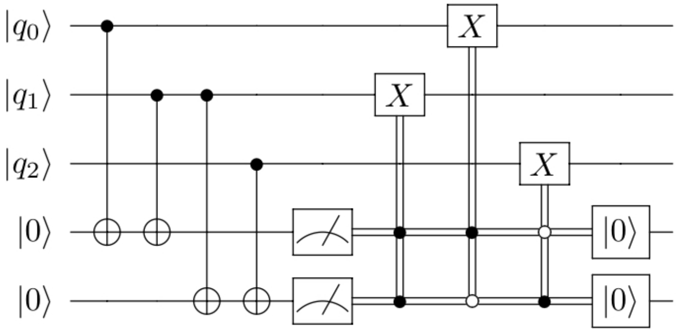


$|q_0\rangle$
$|q_1\rangle$
$|q_2\rangle$
$|0\rangle$
$|0\rangle$
X
X
X
$|0\rangle$
$|0\rangle$

Step 3: measuring parities

- We measure parities twice, so our two backup (ancilla) qubits have the parities $|q_0\rangle \oplus |q_1\rangle$ and $|q_1\rangle \oplus |q_2\rangle$
- If the parities are $\{0, 0\}$, no qubits have flipped
- If they're $\{1, 0\}$, the left one has flipped
- If they're $\{0, 1\}$, the right one has flipped
- If they're $\{1, 1\}$, the middle one has flipped
- We check the parities by measuring both ancilla qubits, so we don't measure our original qubit
- The measurement has an additional benefit: it forces partial bit flips to disappear or become complete bit flips

The circuit

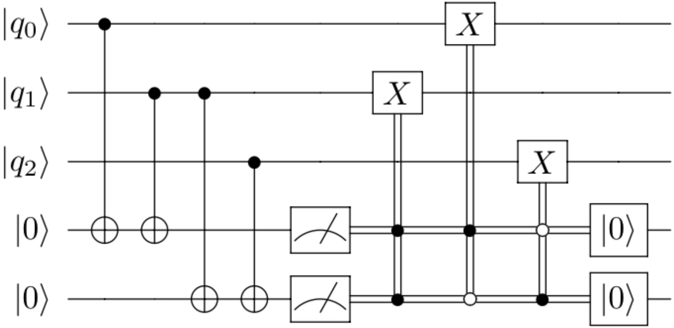


$|q_0\rangle$
$|q_1\rangle$
$|q_2\rangle$
$|0\rangle$
$|0\rangle$
X
X
X
$|0\rangle$
$|0\rangle$

Step 4: correcting the error

- Remember Pauli-X gates act as bit-flips
- We put a Pauli-X on each physical qubit $|q_0\rangle, |q_1\rangle, |q_2\rangle$ and just flip it depending on what the parities tell us
- The X gates are controlled with the two ancilla qubits
- The X gate is classically the NOT gate, so this is actually the CCNOT gate

The circuit

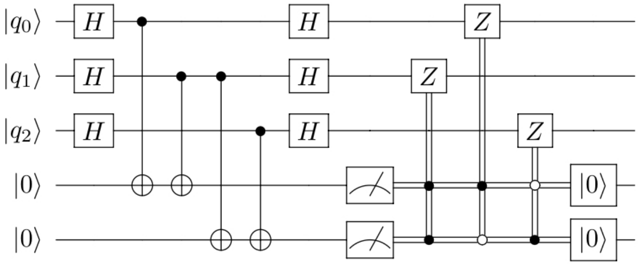


$|q_0\rangle$
$|q_1\rangle$
$|q_2\rangle$
$|0\rangle$
$|0\rangle$
X
X
X
$|0\rangle$
$|0\rangle$

Phase-flip code

- To correct phase-flips, we take advantage of the Hadamard gate
- Phase-flips are errors in the ONB $\{|+\rangle, |-\rangle\}$, which the Hadamard gate converts to $\{|0\rangle, |1\rangle\}$
- We simply convert the ONB with Hadamards, run the algorithm to calculate bit parities, convert back, and use Pauli-Z gates to correct the errors

The phase-flip circuit



Questions?

Nothing can create something all the time due to the laws of quantum mechanics, and it's - it's fascinatingly interesting.

— LAWRENCE M. KRAUSS

Bibliography

Thomas Wong.

Introduction to classical and quantum computing.

https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e3p.pdf, 2022.

Accessed: 10-07-2023.