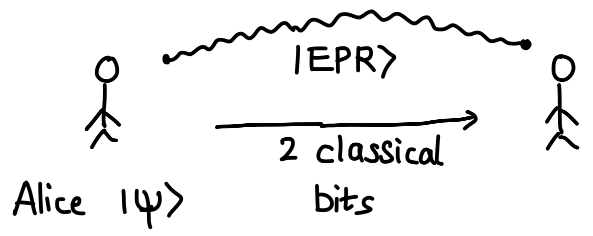
LECTURE 9 September 19^{th}, 2023

Today: Exchanging Quantum Information (contd)

PART II: Fundamental Quantum Algorithms

└ Basics of Quantum Computing

RECAP Quantum Teleportation



Alice has $|\psi\rangle$

Alice & Bob share an EPR pair

They can exchange classical messages

Alice sends Bob 2 bits & Bob gets a perfect copy of $|\psi\rangle$

KEY Idea If Alice performs a local CNOT on $|\psi\rangle$ & her share of the EPR pair, then all three qubits become entangled. If Alice measures, she has performed a distributed CNOT essentially. There might be some errors but Bob can correct them if Alice sends the measurement outcome.

How much information can be encoded in qubits?

n-qubit state

$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$ has 2^n complex amplitudes

On the surface it looks like it contains an exponential amount of information

In Quantum Computing, we want to harness this to our advantage

But as we have seen, we can only get information by measurements which changes the quantum state, so there is a delicate balance here

If we have n qubits, how much classical information can we store?

Can we use n qubits as a "quantum hard drive" to store much more than n classical bits?

Holevo's Theorem says that for information storage quantum bits are not much better than classical bits

Theorem Alice has an m-bit string X that she wants to transmit to Bob. She wants to encode X in some n-qubit state $|\psi_X\rangle$ s.t. Bob can do local operations to try to decode X

Bob only gets X with high probability if $n \geq m$

$\Rightarrow \mathbb{P}[\text{Bob decodes X correctly}] \leq 2^{n-m}$

More to this story

If Alice & Bob share an EPR pair she can send only $\frac{m}{2}$ qubits & Bob can recover with high probability.

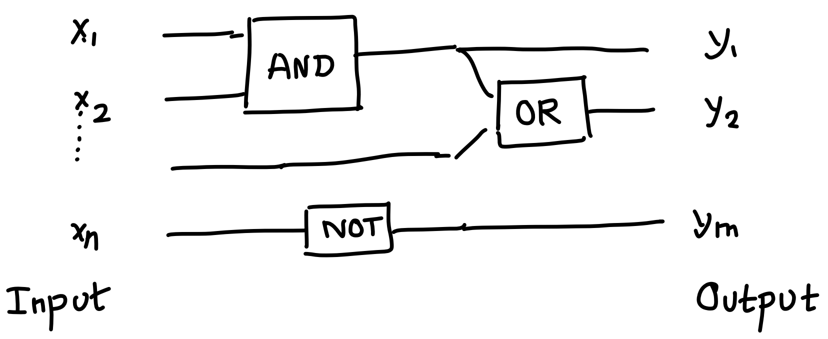
" 1 ebit + 1 qubit $\geq$ 2 classical bit "

This is called superdense coding & very similar to teleportation

Basics of Quantum Computing

First let us start with the basics of classical computing

Classical Circuit C



Computes a function $F: \{0,1\}^n \longrightarrow \{0,1\}^m$ ← Typically just consider $m=1$ since we can output bit by bit

$\qquad\qquad\qquad\qquad\qquad\quad$ n-bits $\qquad\quad$ m-bits

E.g. $x_1, \dots x_n$ = bit representation of a large number

y_1 = if x is prime or not

Our focus will be on efficiency — design circuits with fewest number of gates in particular, how does the # gates scale with n ? Is it n^2 or 2^n ?

\# Gates in a classical circuit roughly corresponds to number of time steps an algorithm takes

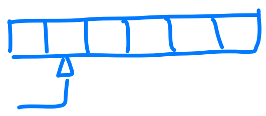
$$\# \text{ Gates} \approx \# \text{ time-steps}$$

How would you implement this as a program or a Turing Machine ?

E.g. python

```
def F(x):
    ___
    ___
    return y
```

OR Turing Machine



FACT: Given python code that computes F in T steps on length n-inputs, one can produce a circuit using {AND, OR, NOT} gates that computes F and has $c_{\text{python}} \cdot T \log T$ gates ?

Shannon's Theorem (1937) Every $F : \{0,1\}^n \to \{0,1\}$ can be computed by a circuit with 2^n gates.

Also, almost all F's need $\geq \frac{2^n}{n}$ gates.

The functions we care about are special and in some cases we only need polynomial in n gates

E.g. shortest path

Probabilistic Computation

· Add a FLIP gate

$\boxed{\text{FLIP}}$ — $\begin{cases} 0 & \text{w.p. } 1/2 \\ 1 & \text{w.p. } 1/2 \end{cases}$ random output

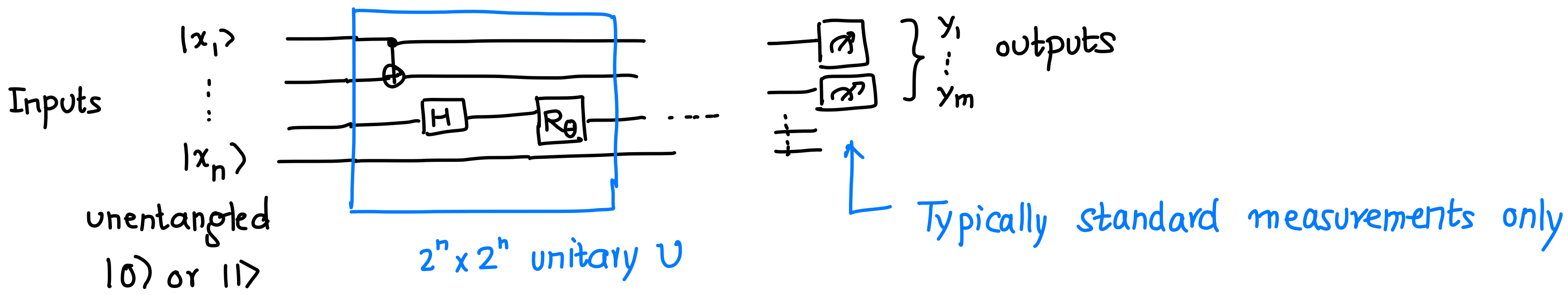
no input

A probabilistic circuit C "computes" $F : \{0,1\}^n \to \{0,1\}$ if

$\forall$ inputs x, $\underset{\text{flips}}{\mathbb{P}}[C(x) \neq F(x)] \leq$ small (say 1%) ← We can reduce the error by repeating

We strongly believe that probabilistic computation does not give exponential savings

Quantum Circuit Model



Quantum circuit C computes $F : \{0,1\}^m \to \{0,1\}$ if

$\forall$ inputs x, $\underset{\text{measurement}}{\mathbb{P}}[C(x) \neq F(x)] \leq$ small

Typically we measure only at the end since we can add extra qubits to defer the measurement → "Principle of Deferred Measurement"
Will be on HW 3

This also is nice since we don't have to deal with mixed states since there are no intermediate measurements

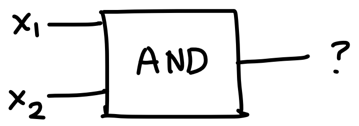
Next few lectures

Q1: Does $\exists\, F$ which quantum circuits can compute much more efficiently than classical ones?

Q2: Can quantum circuits simulate classical circuits?

Q3: Does the exact quantum gates matter?

Let's talk about Q2 first! Can a Q.C. compute



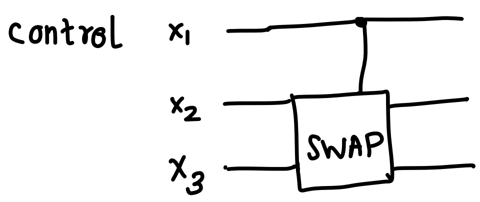
Recall, Q. gates are unitary $UU^{+} = I \Rightarrow U^{-1} = U^{+}$ inverse exists

In particular, all quantum gates are reversible meaning if you know the output you can figure out the input

Not true for AND! True for NOT gate!

This topic of reversible computing was studied by physicists in the 1960s-70s
They were interested in energy efficiency

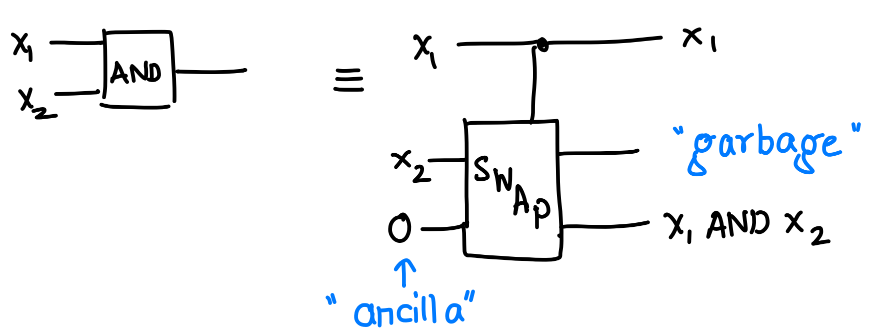
To make AND gate reversible, we need more qubits and one reversible gate CSWAP
(Controlled-SWAP)



SWAPS x_2 and x_3 if $x_1 = 1$
does nothing if $x_1 = 0$

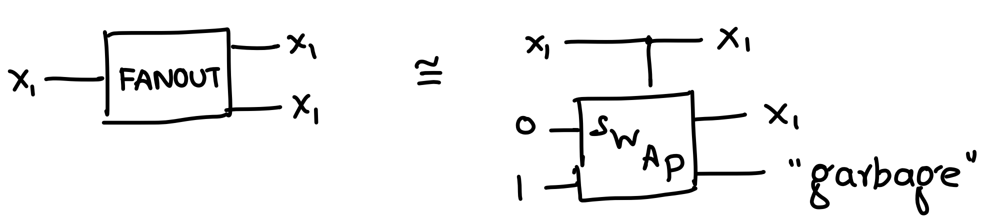
Gives a permutation of $\{|000\rangle, \ldots, |111\rangle\} \Rightarrow$ Unitary gate
It's it own inverse in fact!

How to implement AND gate?



Similarly one can implement OR gate using CSWAP

How to implement FANOUT gate?



Corollary Any classical circuit computing $F: \{0,1\}^n \to \{0,1\}^m$ can be efficiently converted to a reversible (and hence quantum) circuit

$$QC: \{0,1\}^{n+a} \longrightarrow \{0,1\}^{m+g} \qquad n+a = m+g$$

a = # of ancilla bits ← typically initialized to all 0's
g = # of garbage bits

$$(\underbrace{x}_{n}, \underbrace{0,\dots,0}_{a}) \xrightarrow{QC} (\underbrace{F(x)}_{m}, \underbrace{g(x)}_{g})$$

gates in quantum circuit $\leq$ (# gates in classical circuit) · constant

How about probabilistic computing ?

$\boxed{\text{FLIP}}$ — 0/1 random $\quad\cong\quad$ $|0\rangle$ — $\boxed{H}$ — — 0/1 random

Can defer measurement till the end using more ancillas

$\boxed{\text{Summary}}$ Quantum circuits are at least as powerful as probabilistic circuits

$\boxed{\text{NEXT TIME}}$ More on garbage & Q3 whether the exact quantum gate set matter ?