CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Thursday, January 25 - Lecture 3.2

Today's scribe: Daniel

Reading: Chap 1.43, 1.44, Chap 4.41, 4.6 of Nielsen and Chuang.

Agenda:

1. Di Vincenzo's Criteria
2. Generalities on quantum circuits
3. Implementing classical Boolean functions with quantum circuits

1 DiVincenzo's Criteria

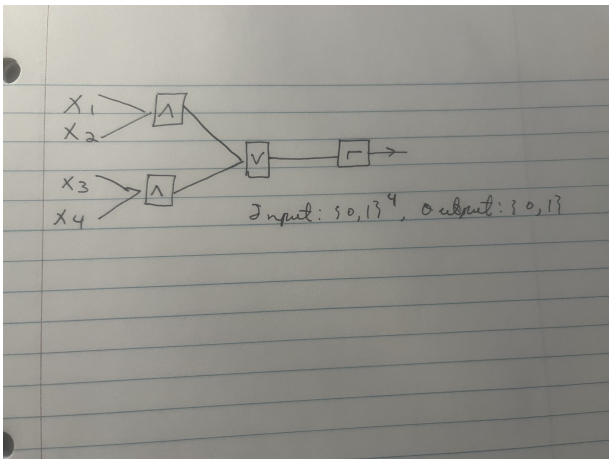
To build a quantum computer we need the following:

1. A scalable physical system with well characterized qubits.
2. The ability to initialize the qubits to some standard input state such as $|00...0\rangle$.
3. Good qubits which are relatively stable. That is, we need the coherence time of the qubits to be longer than the time it takes to implement any computations.
4. A universal set of quantum gates (could be in an encoded sense, i.e. using error correcting codes).
5. The ability to perform measurements in the computational basis.

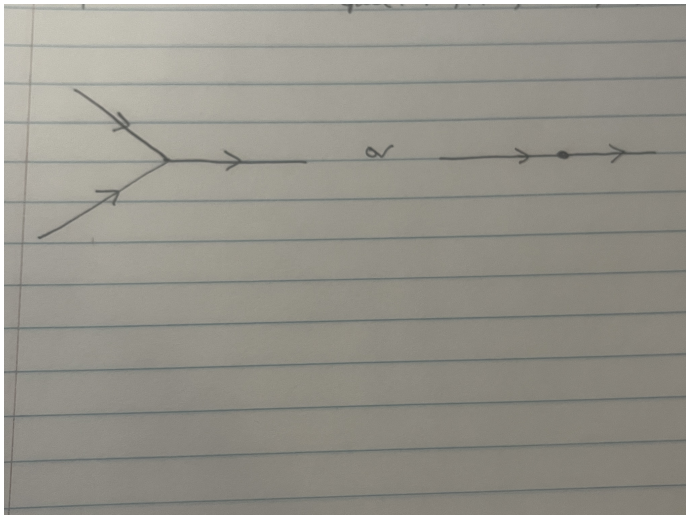
Rather than discussing hardware that may achieve these criteria, in this course we will focus on the abstraction most commonly used to describe a computer that operates according to these criteria: quantum circuits. Today we will focus on the abstract strucure of circuits, as well as show how to encode classical computations inside of quantum circuits. We will discuss what "universality" of a gate set means next time.

2 Quantum Circuits: Analogs of Classical Boolean Circuits

Before we discuss quantum circuits, we should review classical Boolean circuits. A Boolean circuit is something like this:



More precisely, a Boolean circuit over the gate set $\{AND = \wedge, OR = \vee, NOT = \neg\}$ is a directed acyclic graph where every internal vertex looks like:



with the further condition that the vertices with one incoming and one outgoing edge are decorated with the $\neg$ gate and each of the vertices with two incoming edges is decorated either with the $\wedge$ or $\vee$ gate.

A boolean circuit C should be understood as a "factorization" of a function $C : \{0,1\}^n \mapsto \{0,1\}^m$. Here n equals the number of inputs bits and m the number of output bits.

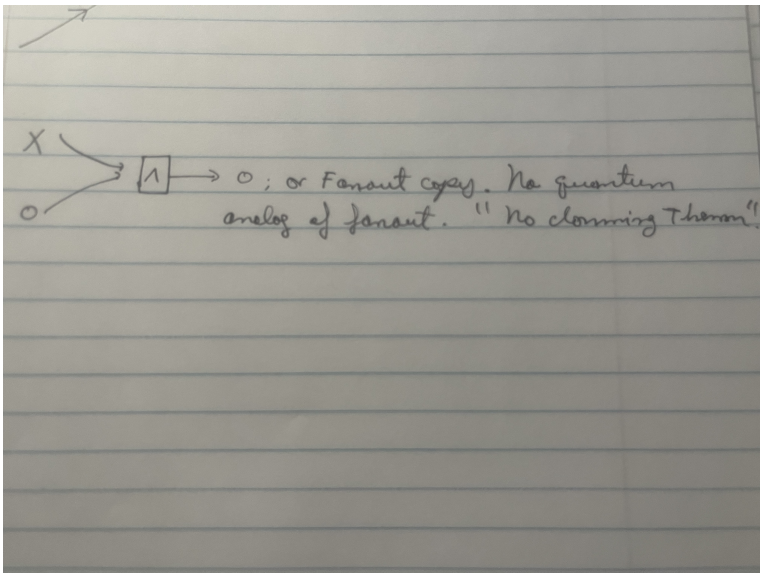
Two key facts:

1. Every Boolean function $F : \{0,1\}^n \mapsto \{0,1\}^m$ can be expressed as a Boolean circuit over $\wedge, \vee, \neg$. However, we also we need to allow additional input bits called ancillas, which we will always set to the initial input value 0.

   For example, this function can't be implemnted without ancillas:

   $$F : \{0,1\} \to \{0,1\}$$
   $$0 \mapsto 0$$
   $$1 \mapsto 0$$

   Instead of ancillas, one could include include fan-out gates. However, since the no-cloning theorem says we can't copy quantum states, it's best not to do this, as it won't be generalizable to quantum circuits. (The notion of ancilla does generalize well, on the other hand.)



2. Given a quantum circuit C, let

$$CSAT(C) = \begin{cases} \text{YES}, & \text{if } C(x) = 1 \text{ for some } x \\ \text{NO}, & \text{otherwise.} \end{cases}$$

CSAT is NP complete. This is called the Cook-Levin Theorem.

Let C be a circuit and x an input to C. Define $EVAL(C,x) = C(x)$.Then $EVAL(C,x)$ is P-complete (under Log-space reduction).

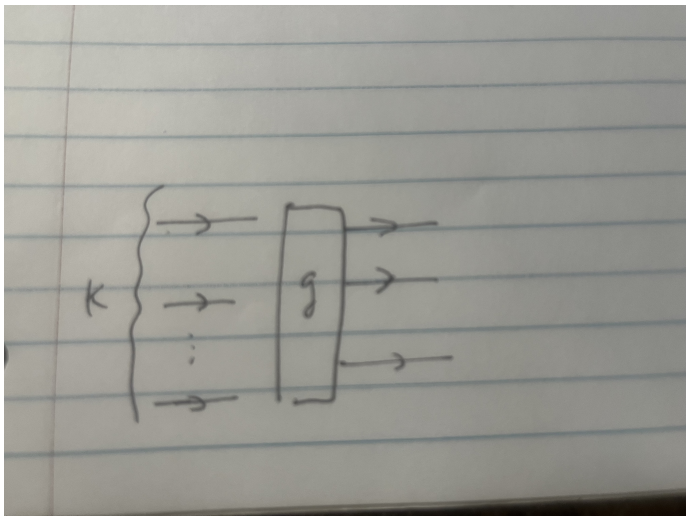
A quantum circuit is like a Boolean circuit with bits replaced with qubits, and Boolean gates replaced with quantum gates. A *quantum gate on k qubits* is a unitary operator $g : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes k}$. To define quantum circuits, we will want to fix a set of quantum gates and only use those to build our circuits. Let's introduce some notation to this end.

Let $U(n)$ be the unitary group of $\mathbb{C}^n$, defined as the set of all unitary operations

$$U(n) = \{U : \mathbb{C}^n \to \mathbb{C}^n \mid U^* = U^{-1}\}.$$

A *gate set* is any set $\mathscr{G} \subseteq \bigsqcup_{k\geq 1} U(2^k)$. Let $\mathscr{G}_k = \mathscr{G} \cap (U(2^k))$ be the set of *k-ary quantum gates* in $\mathscr{G}$.

If $g \in \mathscr{G}_k$ we express this diagrammatically like this:



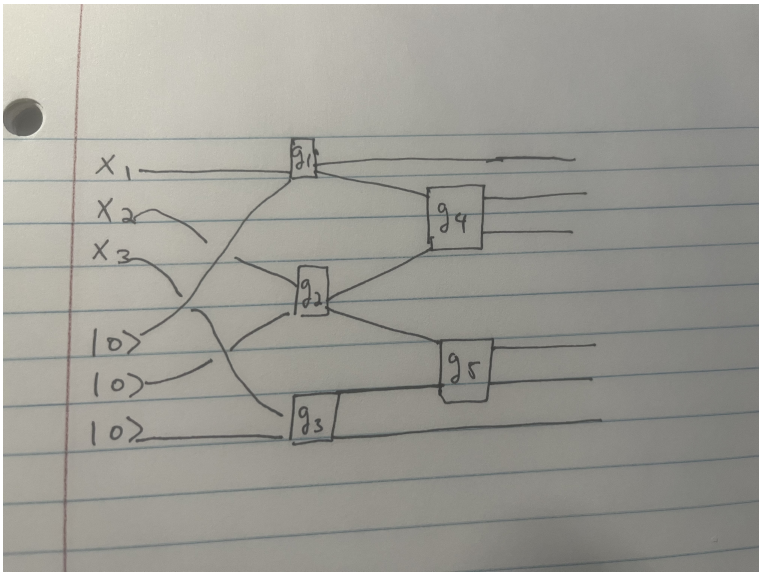
A *quantum circuit over* $\mathscr{G}$ is a directed acyclic graph C such that all internal vertices have input valance = output valence, together with the following data: an ordering of the input edges at each internal vertex, and a labeling of each vertex with incoming valence k by elements of $\mathscr{G}_k$.[1] If C has n incoming edges (and, hence, n outgoing edges) we say that C is a circuit on n *qubits*.

People often restrict to planar circuits, and include SWAP gates in $\mathscr{G}$. If we do this, then the planar structure of the circuit keeps track of which inputs are which for each gate (as is clear in all of our pictures).

Similar to a Boolean circuit, a quantum circuit on n qubits should be thought of as a factorization of a "big" unitary operator on n qubits in term of small operators $U_C : (\mathbb{C}^2)^{\otimes n} \mapsto (\mathbb{C}^2)^{\otimes n}$.

Just as for Boolean circuits, we will want to consider ancillas. A *quantum circuit with an ancilla register* is a quantum circuit where the input qubits are partitioned into 2 subsets called the computational register and the ancilla register. We will always initiate the ancilla qubits to be in the $|0....0\rangle$ state. Here is an example:

$$g_i \in \mathscr{G} \subseteq U(\mathbb{C}^2 \otimes \mathbb{C}^2) \cong U(\mathbb{C}^4)$$



[1]The ordering of the set of edges is necessary since it need not be true that $\mathscr{G}$ is closed under permutation of the tensor factors. For example, it need not be true that $g(|x\rangle \otimes |y\rangle) = g(|y\rangle \otimes |x\rangle)$ for all $g \in \mathscr{G}_2$.

We say this circuit is of depth 2, size 5 (that is, it has 5 gates) and width 6. (If we count the swaps as gates, then it has dpeth 4, actually.)

This concludes the definition of quantum gates with respect to a gate set $\mathscr{G}$. One of the biggest differences between quantum circuits and classical circuits is that all gates must have the same number of inputs as outputs. We impose this requirement because time evolution in a quantum system is always unitary.

Of course, the kinds of things we can compute depend on the choice of $\mathscr{G}$. Moreover, only some choices of $\mathscr{G}$ are realistic. We'll start digging into this now. First, we will show that even though all gates in a quantum circuit are required to be unitary, it is still possible to encode classical Boolean circuits with quantum circuits.

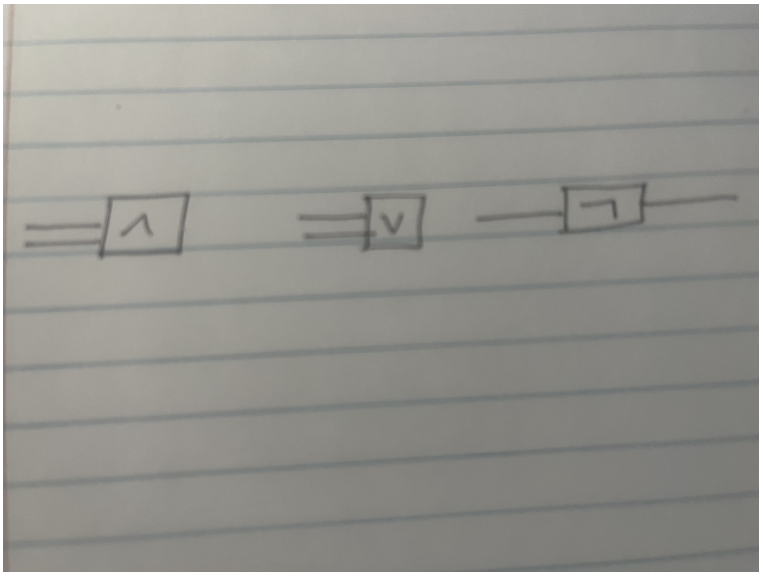
3 Encoding Boolean functions with quantum circuits

Let us say that a quantum circuit *C exactly computes the Boolean function* $F : \{0,1\}^n \mapsto \{0,1\}$ if:

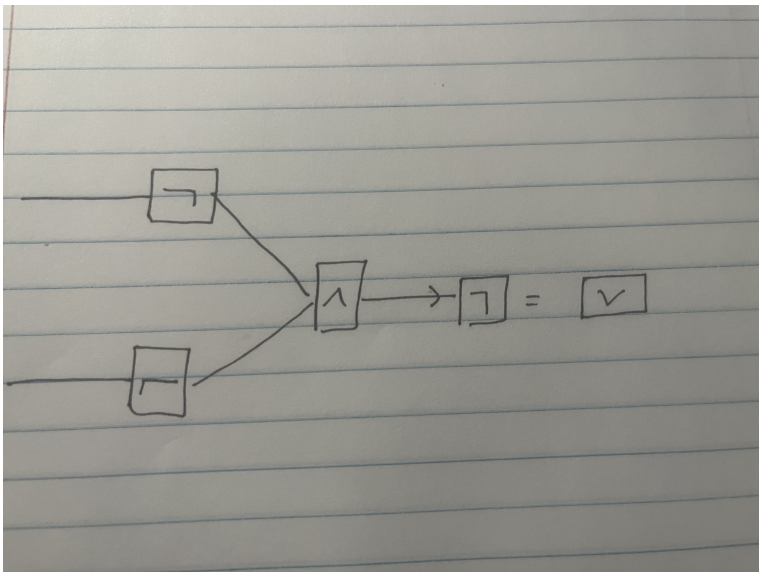
1. C has n computational qubits and any number of ancillas.
2. If $b \in \{0,1\}^n$ then the result of measuring the first qubit of $C(|b\rangle \otimes |0...0\rangle)$ in the computational basis is $F(b)$ with probability 1. (Since we're insisting on getting $F(b)$ with certainty, this is why I call this "exactly computing" F.)

Claim: if $\mathscr{G} = U(2^3)$ then every Boolean function $F : \{0,1\}^n \mapsto \{0,1\}$ can be computed exactly by some circuit over this $\mathscr{G}$.

Proof: Since F can be expressed as a Boolean circuit over $\wedge, \vee, \neg$ it suffices to show that we can compute each of these 3 exactly with a quantum circuit.



In fact, de Morgan's law says



So it suffices just to do $\wedge$ and $\neg$.

The not gate $\neg$ is easy. Abusing notation, define

$$\begin{aligned} \neg : \mathbb{C}^2 &\to \mathbb{C}^2 \\ |0\rangle &\mapsto |1\rangle \\ |1\rangle &\mapsto |0\rangle \end{aligned}$$

(This is also called the Pauli X gate.) You might think that we're done, but recall that we're supposed to show that 3-ary gates are universal, and this is 1-ary gate! But now just let $g_\neg = \neg \otimes Id_{\mathbb{C}^2} \otimes Id_{\mathbb{C}^2}$.

Finally for the $\wedge$ gate we use a technique called unitary dilation. The trick exploits ancillas. Define:

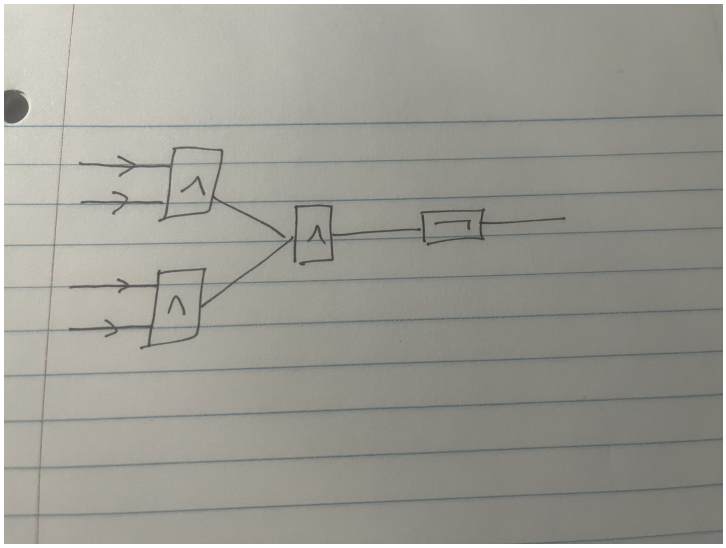
$$g_\wedge : \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \to \mathbb{C}^2 \times \mathbb{C}^2 \otimes \mathbb{C}^2$$

$$|a,b,c\rangle \mapsto |a,b,(a \wedge b) \oplus c\rangle.$$

We can see by direct computation that $g_\wedge \in U(2^3)$ is simply a permutation of the computational basis.

$$\begin{array}{ccc} 0\ 0\ 0 & & 0\ 0\ 0 \\ 0\ 0\ 1 & & 0\ 0\ 1 \\ 0\ 1\ 0 & & 0\ 1\ 0 \\ 0\ 1\ 1 & & 0\ 1\ 1 \\ 1\ 0\ 0 & \mapsto & 1\ 0\ 0 \\ 1\ 0\ 1 & & 1\ 0\ 1 \\ 1\ 1\ 0 & & 1\ 1\ 1 \\ 1\ 1\ 1 & & 1\ 1\ 0 \end{array}$$

Now given a Boolean circuit for F over $\wedge, \neg$, replace each classic and negation gates by their quantum analogs and wire things up "the only way that makes sense." You might do this small Boolean circuit as an example:



□

Note that we actually showed something significantly stronger. We did not need $\mathscr{G} = U(2^3)$ to be *all* 3-ary operations (an uncountably infinite set). We just needed two matrices that are permutations in the computational basis (one for $\neg$ and one for $\wedge$). Next time we will show that there exists a finite gate set that is able to approximate arbitrary unitaries approximately well to arbitrary precision.