Lecture 11 – Revealing XOR-patterns I

Recap: $\forall$ AND/OR/NOT circuit computing some $F: \{0,1\}^n \to \{0,1\}^m$ (often $m=1$), can easily convert to a quantum circuit Q_F that:
- "implements" F on classical inputs
- can also accept superpositions.

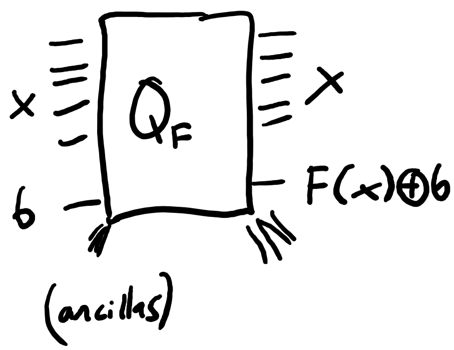
Implements:

$$|x\rangle|b\rangle|00\cdots0\rangle \overset{Q_F}{\longmapsto} |x\rangle|b \oplus F(x)\rangle|00\cdots0\rangle$$

where:
- $\otimes$ signs omitted (standard practice: $|y\rangle|z\rangle$ means $|y\rangle \otimes |z\rangle$)
- $x \in \{0,1\}^n$ ("input register")
- $b \in \{0,1\}^m$ ("output register")
- $\oplus$: bitwise XOR
- $|00\cdots0\rangle$: ancillas $\leftarrow$ we often skip writing them

"sign-implementation trick" (only for $m=1$)

(This is a trick that isn't essential, and in some ways I feel bad about forcing you to remember another trick. But it's just so convenient sometimes....)



Put in $b = |0\rangle$: get out $|F(x)\rangle$

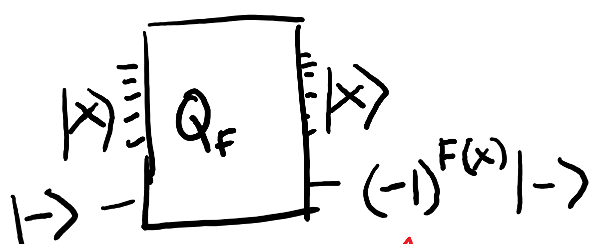
$b = |1\rangle$: $|\neg F(x)\rangle$

$b = |-\rangle$: (!!) ?

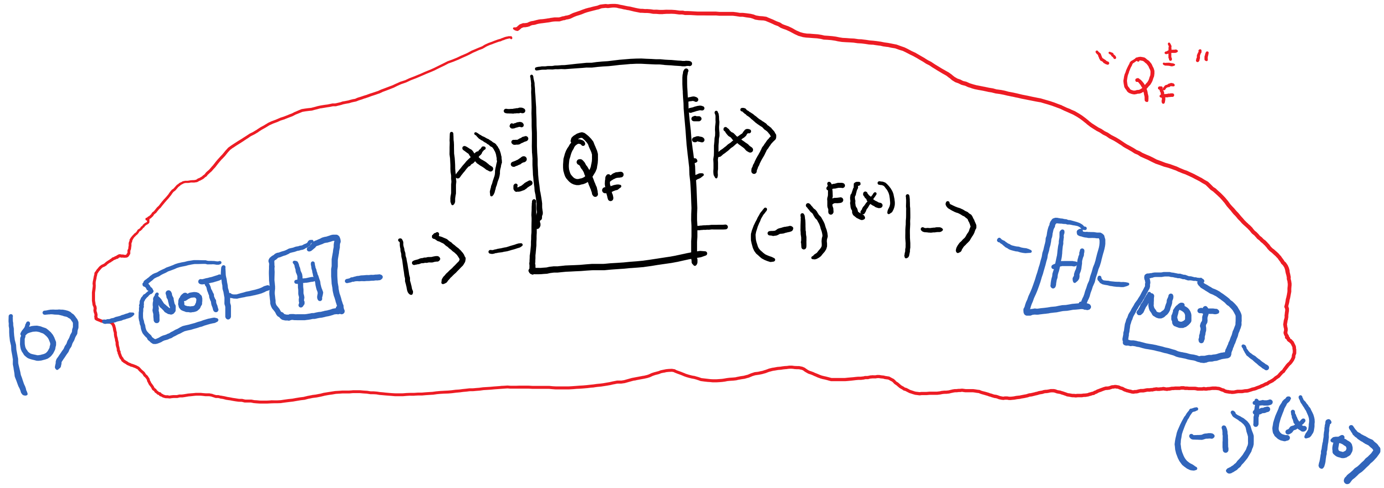
$$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \rightsquigarrow \underbrace{\frac{1}{\sqrt{2}}|F(x)\rangle - \frac{1}{\sqrt{2}}|\neg F(x)\rangle}$$

either $|-\rangle$ if $F(x)=0$
or $-|-\rangle$ if $F(x)=1$

$\therefore$



[The computation is effected "in the phase/sign"]



(Ignoring ancillas) now:

$$|x\rangle \otimes |0\rangle \longmapsto |x\rangle \otimes \left((-1)^{F(x)}|0\rangle\right)$$
$$= (-1)^{F(x)}\,|x\rangle \otimes |0\rangle$$

Just like one more (ignorable) ancilla!

def: $Q_F^{\pm}$ <u>sign-implements</u> $F: \{0,1\}^n \to \{0,1\}$

if $|x\rangle \otimes |00\ldots0\rangle \longmapsto (-1)^{F(x)}|x\rangle \otimes |00\ldots0\rangle$

(ignorable ancillas)

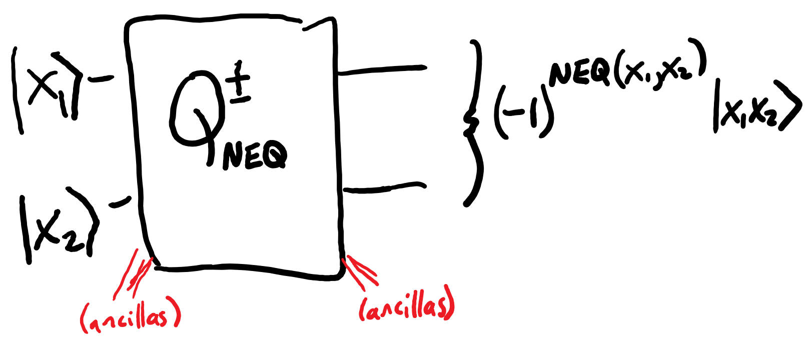
Example: $F : \{0,1\}^2 \to \{0,1\}$ is the "not-equals function", NEQ.

x_1	x_2	$NEQ(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

classical circuit C:

$(x_1 \text{ AND } \neg x_2) \text{ OR } (\neg x_1 \text{ OR } x_2)$

Sign-implementation:



(So you can happily plug in the 4 classical inputs:)

$$|00\rangle \mapsto +|00\rangle$$
$$|01\rangle \mapsto -|01\rangle$$
$$|10\rangle \mapsto -|01\rangle$$
$$|11\rangle \mapsto +|11\rangle$$

Power of Quantum: can plug in superpositions!

e.g. $\frac{1}{2}\left(|00\rangle - |01\rangle - |10\rangle + |11\rangle\right)$ $\circledast$

$\| $ claim

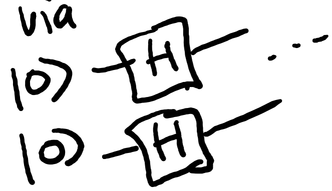
(Might first ask, "How do I even prepare that state?")

$|+\rangle \otimes |+\rangle$

$\|$

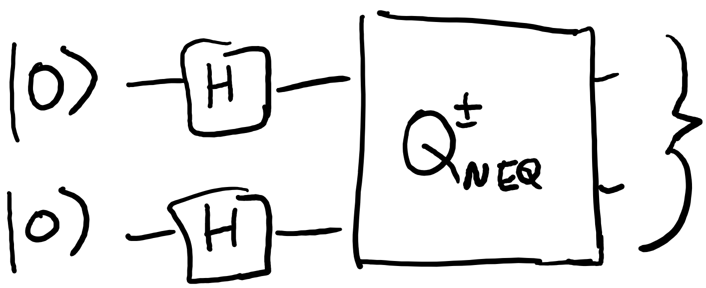
$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$

build via



(Gives $\frac{1}{2} \cdot$ ("foil it"): get all 2-bit strings, as in $\circledast$)

By linearity...



$\frac{1}{2}\left(|00\rangle - |01\rangle - |10\rangle + |11\rangle\right)$

(as always, maybe with unentangled ancillas $\otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \cdots$ tacked on everywhere at end)

COOL! All $4 = 2^2$ answers $(-1)^{NEQ(x)}$, $x \in \{0,1\}^2$
encoded in state's amplitudes —
but we only used $Q^{\pm}_{NEQ}$ 1 time!

COOL? We can't just "look at state" to learn these 4 answers.

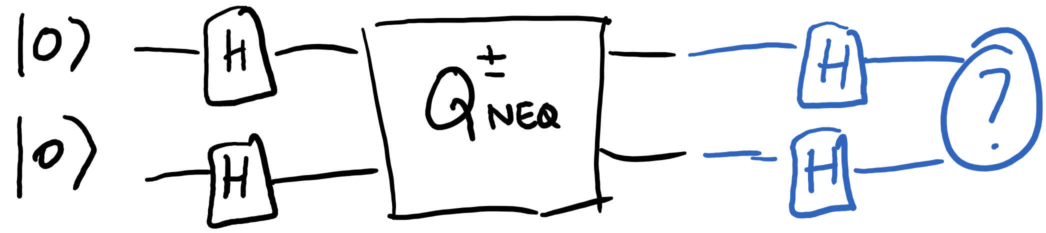
(This is a fundamental critique of the 10^{500} parallel worlds intuition: yes, if $F:\{0,1\}^{500} \to \{0,1\}$, you can cook up 500 photons in superposition, run thru $Q^{\pm}_F$, "get" all 2^{500} answers into the state. But how to "connect" those amplitudes to get anything useful out?!)

Measure now? Naw, just gives a random $x \in \{0,1\}^2$.

Had we used Q_F: $\frac{1}{2}\big(|00,\underset{\uparrow}{0}\rangle + |01,1\rangle + |10,1\rangle + |11,1\rangle\big)$
(NEQ(0,0))

Measure: $\to$ a random $(x, F(x))$ pair. :(

(Also pointless. Could have gotten that classically by picking a rand. x & computing F via 1 application of C.)



ROTATE — COMPUTE — ROTATE!

(a unitary/rotation in $\mathbb{C}^4$)

(We didn't fully do Simon's mantra yet!)

"Hadamarding each qubit": $H \otimes H \otimes \cdots$

$= H^{\otimes n}$ = "Hadamard transform"

= "Boolean Fourier transform"

Final State @ ?: $\frac{1}{2}\left(|{+}{+}\rangle - |{+}{-}\rangle - |{-}{+}\rangle + |{+}{+}\rangle\right)$

$H|0\rangle = |+\rangle$

$H|1\rangle = |-\rangle$

$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$

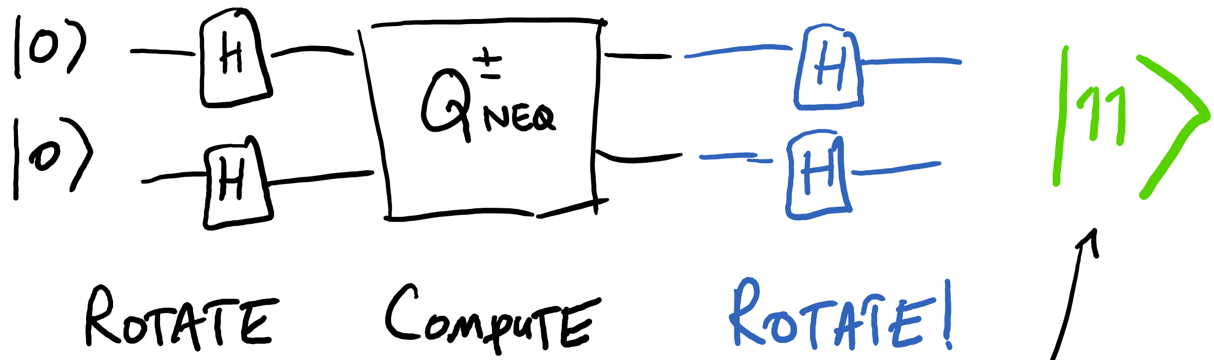
$=$? (Looks like a big mess...)

Start by figuring out amplitude on $|11\rangle$

$$\frac{1}{2}\left(\frac{1}{\sqrt{2}}\cdot\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\left(-\frac{1}{\sqrt{2}}\right) - \left(-\frac{1}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}}\right) + \left(\frac{1}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}}\right)\right)$$

$= 1.$!!

$\therefore$ all other amplitudes must be 0!



Lots of "interference" occurred!

↓

sign-cancellation

Lost the 4 "answers", but created simple state due to "pattern" in the answers.

We'll see: got "100% amplitude" on $|11\rangle$ "because of" an "XOR-pattern" in truth-table of NEQ.

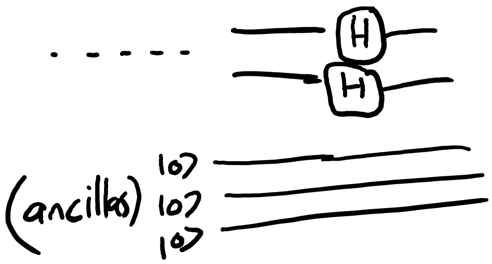
Actually... NEQ = XOR!

00	0
01	1
10	1
11	0

(This "make superpos. of all inputs", "get" all answers, "interfere" them with a discrete Fourier transform, detect pattern, is the true power of Q.C.)

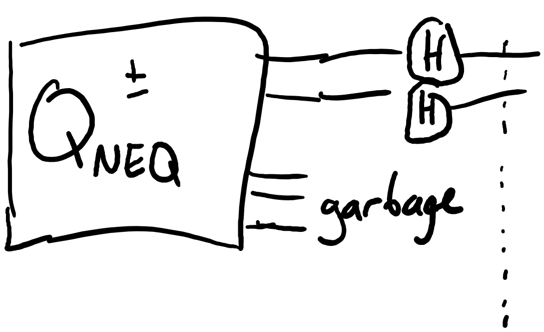
Aside on ancillas/garbage

Actually:



(The last layer of Hadamards doesn't touch ancillas. They really are unentangled, Bob could have 'em on Jupiter, they effect nothing.)

But suppose $Q^{\pm}_{NEQ}$ produced garbage.



$$\frac{1}{2}\Big(|{+}{+}, garb_0\rangle - |{+}{-}, garb_1\rangle - |{-}{+}, garb_2\rangle + |{-}{-}, garb_3\rangle\Big)$$

$\downarrow$

expand out: breaks nice interference! ☹

e.g. amplitude on $|11, stuff\rangle$ is...

$$\frac{1}{4}|11, garb_0\rangle + \frac{1}{4}|11, garb_1\rangle + \frac{1}{4}|11, garb_2\rangle + \frac{1}{4}|11, garb_3\rangle$$

(No "positive interference", still lots of amplitude left....)

In general...

Say $F : \{0,1\}^n \to \{0,1\}$ computed by classical circuit C.

Sign-implemented by $Q_F^{\pm}$.

$$|x\rangle \otimes |00\cdots 0\rangle \xmapsto[Q_F^{\pm}]{} (-1)^{F(x)} |x\rangle \otimes |00\cdots 0\rangle$$

"Rotate 1": Initialize n $|0\rangle$'s, put thru $H^{\otimes n}$.

(Equiv, initialize n $|+\rangle$'s. Easy to do physically, since it's all unentangled.)

$$\rightsquigarrow \quad |+\rangle \otimes |+\rangle \otimes |+\rangle \otimes \cdots = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \cdots$$

$$= \frac{1}{2^{n/2}} \{\text{all combinations}\}$$

$$= \frac{1}{\sqrt{N}} \cdot \sum_{x \in \{0,1\}^n} |x\rangle,$$

"N" denotes 2^n.

def: The <u>uniform superposition</u> on n-bit strings:

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

(Rem: When we study number-theoretic algs. like Shor's, binary strings will be #'s in base 2. Unif. superpos: $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |\text{base-2 rep. of } x\rangle$.)

"Compute"

Plugging unif. superposition into $Q_F^{\pm}$ yields

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} (-1)^{F(x)} |x\rangle. \quad (*)$$

(Going to stop stressing about this "normalizing constant".)

Notation: $f(x) = (-1)^{F(x)}$.

So $f : \{0,1\}^n \to \{\pm 1\}$

(Like the truth table of F, but $0 \mapsto +1$, $1 \mapsto -1$.)

So $(*) = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} f(x) |x\rangle$

$$= \frac{1}{\sqrt{N}} \begin{bmatrix} f(00\ldots0) \\ f(00\ldots1) \\ \vdots \\ \vdots \\ f(11\ldots1) \end{bmatrix}$$

(Remember this "load up your enormous data into a vector" viewpoint from lecture #2?) →

"Rotate 2" : $H^{\otimes n}$ = Boolean Four. Transf. again

[Much more on the Fourier theory next lecture.]

$$H^{\otimes n}\left(\frac{1}{\sqrt{N}}\sum_x f(x)|x\rangle\right) = \ldots ?$$ (a big mess?)

$$= \frac{1}{\sqrt{N}}\sum_x f(x)\underbrace{H^{\otimes n}|x\rangle}_{\text{what's that?}} = \ldots = \frac{1}{\sqrt{N}}\sum_s \boxed{?}\,|s\rangle$$

what's that?

(we know it if $|x\rangle = |00\cdots0\rangle$, it's unif superpos. C.f. HW3, #6.)

this number will be called "$\hat{f}(s)$"

Let's do example... what is

$$H^{\otimes 3}|110\rangle ?$$

It's $|--+\rangle = \frac{1}{\sqrt{8}}(|0\rangle - |1\rangle)\otimes(|0\rangle - |1\rangle)\otimes(|0\rangle + |1\rangle)$

$$= \frac{1}{\sqrt{8}}\sum_{s\in\{0,1\}^3}(\pm)|s\rangle$$

what's the pattern?

$$(|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle) \qquad (x = 110)$$

What is sign on $|s\rangle = |000\rangle$? $+$

$|001\rangle$ $\quad + \cdot + \cdot + = +$

$|010\rangle$ $\quad + \cdot - \cdot + = -$

$\vdots$

$|111\rangle$ $\quad - \cdot - \cdot + = +$

In general, on $|s\rangle$:

product of signs $i = 1 \dots n$

If $s_i = 0$, sign always $+$.

If $s_i = 1$, sign is $-$ iff $x_i = 1$.

$\therefore$ sign on $|s\rangle$ in $H^{\otimes n}|x\rangle$ is

$$\prod_{i : s_i = 1} (-1)^{x_i} = (-1)^{\sum_{i : s_i = 1} x_i \pmod 2}$$

$$= (-1)^{XOR_s(x)},$$

where $XOR_s(x) = $ XOR of bits x_i where $s_i = 1$

(i.e. $i \in$ SetIndicatedBy s)

(In conclusion...)

thm: $H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{s \in \{0,1\}^n} (-1)^{XOR_s(x)} |s\rangle$

(Time for some math magic!)

Note: $XOR_s(x) = \sum_{i : s_i = 1} x_i \mod 2$

$= \sum_{i=1}^{n} s_i x_i \mod 2$ (symmetric!)

$= XOR_x(s)$ (!)

(Plug into thm, then switch letters "x", "s".)

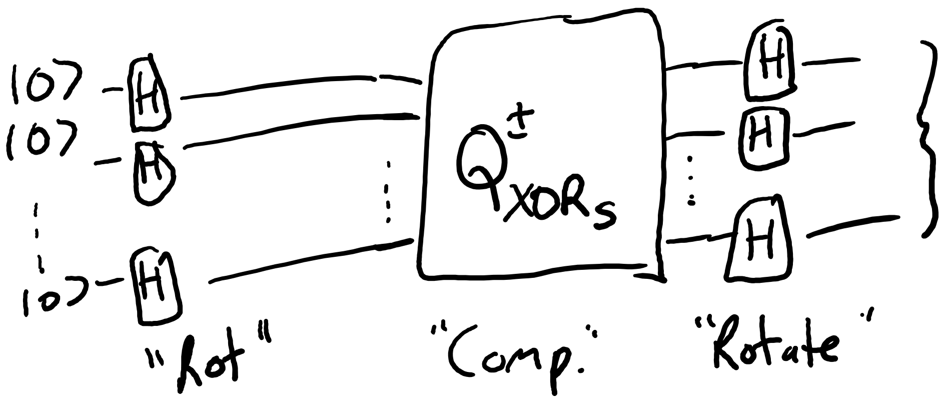
cor: $H^{\otimes n} |s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} (-1)^{XOR_s(x)} |x\rangle$

(But $H = H^{-1}$, so $(H^{\otimes n})^{-1} = H^{\otimes n}$. So...)

cor: $H^{\otimes n}\left(\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} (-1)^{XOR_s(x)} |x\rangle \right) = |s\rangle$

$$H^{\otimes n}\underbrace{\left(\frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}(-1)^{XOR_s(x)}|x\rangle\right)}_{\frac{1}{\sqrt{N}}\sum_x f(x)|x\rangle \text{ for } F=XOR_s} = |s\rangle$$

i.e., output of



(Not a superposition!)

Now measuring reveals "s" w. prob. 100%!

Explains $NEQ(x_1,x_2)$ example. $NEQ(x_1,x_2)=XOR_s(x_1,x_2)$ for $s=11$ (XOR both bits)

(After the "Compute" step, state ampls. encode the $(\pm)$ XOR_s truth table. Final Hadamard transform interference perfectly reveals this "XOR-pattern" in the data!)

(What we've seen here is basically a quantum algorithm due to Bernstein & U. Vazirani in 1993.)

(Imagine like this: someone hands you a chip that implems the Boolean fcn XOR_s, but you don't know s. How to discover it thru I/O behavior?)

(If you can only put in classical inputs x, not hard to see you need to feed in $\underline{n}$ diff. inputs. (see H.W.) But... if the chip accepts superpositions, only need to feed in $\underline{\underline{1}}$ input — the unif. superpos. — and you can learn s with 100% accuracy!)