Excursions in Computing Science: Book 11d. Forces and Invariants Part VI. Quantum Computing

T. H. Merrett*
McGill University, Montreal, Canada

August 2, 2024

38. Quantum Computing. A quantum computer is a different piece of hardware from a classical computer, operating on different principles. We start with the principles because they are based on now-familiar mathematics. Quantum computing also emphasizes some aspects of quantum physics which are still unfamiliar to us. We will hint at some hardware in Note 47.

A classical bit has two values which we will call `F` and `T` to make a clear distinction from the numbers that follow.

A quantum bit, or *qubit*, represents these two values as the two basis vectors of a two-dimensional space, $(1,0)^{\mathrm{T}}$ and $(0,1)^{\mathrm{T}}$, respectively.

A classical boolean operation, such as **not**, maps `F` into `T` and `T` into `F`. It can be written as a matrix with only 0 s and 1s

$$\overset{\textbf{cnot}}{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}} \begin{pmatrix} \mathtt{F} \\ \mathtt{T} \end{pmatrix} = \begin{pmatrix} \mathtt{T} \\ \mathtt{F} \end{pmatrix}$$

We can use the same matrix to represent the corresponding quantum *gate*, only now what it operates on are the two vectors

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \overset{\mathtt{T}}{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} = \overset{\mathtt{F}}{\begin{pmatrix} 0 \\ 1 \end{pmatrix}} \qquad \text{and} \qquad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \overset{\mathtt{F}}{\begin{pmatrix} 0 \\ 1 \end{pmatrix}} = \overset{\mathtt{T}}{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}$$

I've labelled the qubits with their corresponding bit values for this case, just to emphasize the swap that the matrix performs.

But now we're not restricted just to 0 and 1 in the matrix. We are allowed any complex numbers. (Well, we'll learn about some restrictions.)

Thus, for example, the other two Pauli matrices may now appear

$$\begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} & -i \\ i & \end{pmatrix}$$

An important gate is the matrix that connects two of these Pauli matrices

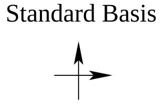
$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$

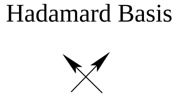
This is the *Hadamard* gate and we know it works this way because it is the reflection in the $22\frac{1}{2}^{\mathrm{o}}$ line. The **not** matrix is the reflection in the 45^{o} line and the other Pauli matrix is the reflection in the horizontal line. The Hadamard reflection of the 45^{o} line is the horizontal line.

The Hadamard gate also introduces into computing the central quantum principle of *superposition.* It allows a qubit to be a superposition of `F` and `T`

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{and} \qquad \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

It introduces a new basis for the qubit space which is sufficiently important to have a name and a notation.





It is handy to have a new notation for these vectors, which relate them back to the classical `F` and `T`. We let $\mid 0>$ stand for `F` and $\mid 1>$ stand for `T`.

$$\mid 0 >= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \mid 1 >= \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Now we can go back to expressing a gate as a single equation.

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} \mid 0 > \\ \mid 1 > \end{pmatrix} = \begin{pmatrix} \mid 1 > \\ \mid 0 > \end{pmatrix}$$

We can also write the Hadamard basis as $\mid + >$ and $\mid - >$

$$\mid + >= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \mid - >= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

But we must be careful. Check this out:

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} \mid + > \\ \mid - > \end{pmatrix} = \begin{pmatrix} \mid + > \\ - \mid - > \end{pmatrix}$$

So the single-equation trick works in the standard basis but not in other bases.

(What must we change in the above to make a correct matrix equation for "$(\mid + >, \mid - >)^{\mathrm{T}}$"?)

It is going to be better to switch to a notation using algebraic combinations of $\mid 0 >$ and $\mid 1 >$. Thus

$$\mathbf{not}(a \mid 0 > + b \mid 1 >) = a \mid 1 > + b \mid 0 >$$

So, in particular for the standard basis

$$\mathbf{not} \mid 0 >= \mid 1 > \qquad \mathbf{not} \mid 1 >= \mid 0 >$$

and for the Hadamard basis

$$\begin{aligned} \mathbf{not} \mid + > &= \mathbf{not}(\frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >)) = \frac{1}{\sqrt{2}}(\mid 1 > + \mid 0 >) = \mid + > \\ \mathbf{not} \mid - > &= \mathbf{not}(\frac{1}{\sqrt{2}}(\mid 0 > - \mid 1 >)) = \frac{1}{\sqrt{2}}(\mid 1 > - \mid 0 >) = - \mid - > \end{aligned}$$

Other unary gates change phase. Calling the Pauli matrices

$$\begin{array}{ccc} X & Y & Z \\ \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} & \begin{pmatrix} & -i \\ i & \end{pmatrix} & \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \end{array}$$

we get the possibilities

$$e^{i\phi Z} = \begin{pmatrix} e^{i\phi} & \\ & e^{-i\phi} \end{pmatrix} \quad \text{and} \quad e^{i\phi X} = \begin{pmatrix} \cos\phi & -i\sin\phi \\ i\sin\phi & \cos\phi \end{pmatrix}$$

as well as the *phase gate*

$$\sqrt{Z} = \begin{pmatrix} 1 & \\ & i \end{pmatrix}$$

Binary gates. Quantum operators, we have learned, must be *unitary* (see, e.g., Note 11 in Part II)

$$UU^{\dagger} = I$$

which requires at least that they are reversible and, indeed, effectively their own inverse[1]. This is certainly true of reflections $FF = 1$, so for the Pauli matrices in particular

$$XX^{\dagger} = I \quad YY^{\dagger} = I \quad ZZ^{\dagger} = I$$

[1] Unitary matrices are so called because their eigenvalues all lie on the unit circle, $e^{i\phi}$, as is necessary for $UU^{\dagger} = I$ in the coordinate system that diagonalizes U and $U^{\dagger}$. Unitary operators conserve the norm (length) of the states.

where X and Z are symmetric, $X^\dagger = X$ and $Z^\dagger = Z$, and where Y is Hermitian, $Y^\dagger = Y$. Check that this is true for all the preceding unary gates.

The only partially (self-) reversible binary Boolean operator is **xor**. But it needs a second output: one of its inputs will do. This way we get a square matrix for the two bits x and y ($\oplus$ is short for **xor**)

$$\begin{array}{ccccc} \textbf{xor} & & x\ y & & x\ x \oplus y \\ \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{pmatrix} & & \begin{pmatrix} \texttt{FF} \\ \texttt{FT} \\ \texttt{TF} \\ \texttt{TT} \end{pmatrix} & = & \begin{pmatrix} \texttt{FF} \\ \texttt{FT} \\ \texttt{TT} \\ \texttt{TF} \end{pmatrix} \end{array}$$

We note that $\texttt{F}$ **xor** $y = y$ but $\texttt{T}$ **xor** $y =$ **not** y. We've kept the input x as the other output, so that **xor** can be applied again to restore the original input.

If we had kept y as the other output the matrix would have been

$$\begin{array}{ccccc} \textbf{xor} & & x\ y & & x \oplus y\ y \\ \begin{pmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{pmatrix} & & \begin{pmatrix} \texttt{FF} \\ \texttt{FT} \\ \texttt{TF} \\ \texttt{TT} \end{pmatrix} & = & \begin{pmatrix} \texttt{FF} \\ \texttt{TT} \\ \texttt{TF} \\ \texttt{FT} \end{pmatrix} \end{array}$$

For the quantum gate, in the standard basis, which matrix we get depends on the convention we use for the tensor product.

With

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \overleftarrow{\times} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$$

$$\texttt{FF} \text{ is } \begin{pmatrix} 1 \\ 0 \end{pmatrix} \overleftarrow{\times} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \texttt{FT} \text{ is } \begin{pmatrix} 1 \\ 0 \end{pmatrix} \overleftarrow{\times} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

etc., so

$$\begin{array}{cccccccccc} & \texttt{FF} & \texttt{FT} & \texttt{TF} & \texttt{TT} & & \texttt{FF} & \texttt{FT} & \texttt{TT} & \texttt{TF} \\ \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & = & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{array}$$

With

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \overrightarrow{\times} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$$

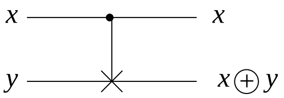
we would need

$$\begin{pmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{pmatrix}$$

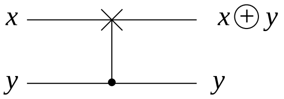
Clearly the **xor** gate is reversible and is its own inverse.

Note that it can also be thought of as a *controlled-not* (**cnot**) gate: apply **not** to the "target" bit but only if the "control" bit is $\texttt{T}$. We can represent this schematically

$$\left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)$$



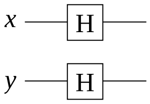
$$\left(\begin{array}{cc|cc} 1 & & & \\ & & & 1 \\ \hline & & 1 & \\ & 1 & & \end{array}\right)$$



So far, the matrices have only 0s or 1s, and so apply equally to bits or to qubits.

But for qubits, if we look at the Hadamard basis, we find more strangeness.

First, let's apply the Hadamard transformation to each of two bits.



$$(H \otimes H)(\mid x > \otimes \mid y >) = (H \mid x >) \otimes (H \mid y >)$$

When we free ourselves from matrices we need no longer worry about the direction of the tensor product, so we can write it symmetrically, $\otimes$. A tensor product of operators applied to a tensor product of states (vectors) just applies each operator to the corresponding state. Thus

$$\begin{aligned}(H \otimes H)(\mid 0 > \otimes \mid 0 >) &= \frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >) \otimes \frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >) \\ &= \frac{1}{2}(\mid 0 >\mid 0 > + \mid 0 >\mid 1 > + \mid 1 >\mid 0 > + \mid 1 >\mid 1 >) \\ &= \frac{1}{2}(\mid 00 > + \mid 01 > + \mid 10 > + \mid 11 >)\end{aligned}$$

where we can abbreviate $\mid 0 > \otimes \mid 0 >=\mid 0 >\mid 0 >=\mid 00 >$, etc. We can even abbreviate further by going from binary to decimal so that

$$H \otimes H \mid 00 >= \frac{1}{2}(\mid 0 > + \mid 1 > + \mid 2 > + \mid 3 >)$$

(I've left the left-hand vector in binary to remind us that there are 2 qubits. There are other ways to disambiguate, such as subscript Bs and Ds.)

The other three results, with abbreviations, are

$$\begin{aligned}(H \otimes H)(\mid 01 >) &= \frac{1}{2}(\mid 00 > - \mid 01 > + \mid 10 > - \mid 11 >) \\ (H \otimes H)(\mid 10 >) &= \frac{1}{2}(\mid 00 > + \mid 01 > - \mid 10 > - \mid 11 >) \\ (H \otimes H)(\mid 11 >) &= \frac{1}{2}(\mid 00 > - \mid 01 > - \mid 10 > + \mid 11 >)\end{aligned}$$

Now look at what **cnot** does in the Hadamard basis.

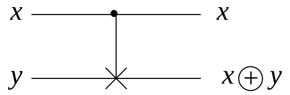
$$\begin{aligned}\textbf{cnot} \mid ++ > &= \textbf{cnot}\ \ H \otimes H \mid 00 > \\ &= \textbf{cnot}\ \ \frac{1}{2}(\mid 00 > + \mid 01 > + \mid 10 > + \mid 11 >) \\ &= \frac{1}{2}(\mid 00 > + \mid 01 > + \mid 11 > + \mid 10 >) \\ &= \mid ++ >\end{aligned}$$

but

$$\begin{aligned}\textbf{cnot} \mid +- > &= \mid -- > \\ \textbf{cnot} \mid -+ > &= \mid -+ > \\ \textbf{cnot} \mid -- > &= \mid +- >\end{aligned}$$

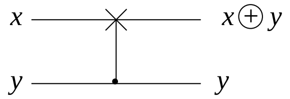
So if **cnot** is

$$\left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)$$



in the standard basis, it is

$$\left(\begin{array}{cc|cc} 1 & & & \\ & & & 1 \\ \hline & & 1 & \\ & 1 & & \end{array}\right)$$



in the Hadamard basis.

It is not always clear which is the (unchanged) "control" qubit for controlled-not, or even if there *is* a "control" qubit.

The Hadamard transformation applied repeatedly to any number of qubits is called the *Walsh* or the *Walsh-Hadamard* transform, and a short way of writing it is $W = H^{\otimes n}$.

Applied to $\mid 0 >$ (decimal 0 here) it gives the superposition of all possible states.

$$W \mid 0 >= (H \otimes \cdots \otimes H) \mid 0 \cdots 0 >= \frac{1}{\sqrt{2^n}}(\mid 0 > + \mid 1 > + \cdots + \mid 2^n - 1 >)$$

This is often a useful state for starting a quantum computation.

Superposition is the first radical departure quantum physics shows from classical physics. Controlled-not shows the second: *entanglement* among two or more states.

$$\begin{aligned}\textbf{cnot}\left(\frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >) \otimes \mid 1 >\right) &= \textbf{cnot}\left(\frac{1}{\sqrt{2}}(\mid 01 > + \mid 11 >)\right) \\ &= \frac{1}{\sqrt{2}}(\mid 01 > + \mid 10 >) \\ &= ? \otimes ?\end{aligned}$$

Here we start with two *separable* states: the left qubit is in state $H \mid 0 >$, the right qubit is in state $\mid 1 >$. The combined initial state is the tensor product of the two.

After **cnot** the state of the two bits is no longer separable. It cannot be written as a tensor product of two separate states.

A physical example of the entangled state $(\mid 01 > + \mid 10 >)/\sqrt{2}$ is the result of the emission by a spin-0 atom of two spin-1 photons. Conservation of angular momentum says that the two photons must have opposite spins, either spin-up and spin-down ($\mid 01 >$) or spin-down and spin-up ($\mid 10 >$). If we don't know which of these two possibilities results, but each is equally likely, then we have the entangled state $(\mid 01 > + \mid 10 >)/\sqrt{2}$.

With enough qubits, there are many more entangled states than pure states. For two qubits a pure state has the form

$$(\alpha \mid 0 > +\beta \mid 1 >) \otimes (\gamma \mid 0 > +\delta \mid 1 >)$$

while an arbitrary state, either pure or entangled, has the form

$$a \mid 00 > +b \mid 01 > +c \mid 10 > +d \mid 11 >$$

These each have four parameters and so occupy 4-dimensional space—the same. But for normalized states $\alpha^2+\beta^2 = 1$, $\gamma^2+\delta^2 = 1$ and $a^2+b^2+c^2+d^2 = 1$ and the pure states span only two dimensions, while all the states fill three dimensions.

For three qubits the gap is apparent even if we do not normalize. Pure states occupy 2+2+2=6 dimensions and entangled states make up the difference to $2\times2\times2 = 8$ dimensions. (These numbers are 3 and 7 if the states are normalized.)

The **cnot** operation which entangled two qubits, one initially in the Hadamard state $\mid + >$and the other initially $\mid 1 >$, is reversible. So, applied again, it can disentangle the result back to the pure state.

Another way to get a pure state from an entangled one is to *measure* it.

If we measure the right bit of $(\mid 01 > + \mid 10 >)/\sqrt{2}$ we "collapse" the state to one, $\mid 01 >$, or the other, $\mid 10 >$, of its components, each with probability $(1/\sqrt{2})^2 = 1/2$ of getting a 1 or a 0 as the result, respectively.

The respective new state after the measurement is

$$\mid 01 > = \mid 0 > \otimes \mid 1 >$$

or

$$\mid 10 > = \mid 1 > \otimes \mid 0 >$$

and is pure.

Measurement. This example of measurement is oversimplified, It assumes that the measurement is being made in the standard basis, $\mid 0 >$ and $\mid 1 >$. That is, the measurement operator is somehow aligned along the $(1,0)^{\mathrm{T}}$ and the $(0,1)^{\mathrm{T}}$ axes—say a polarizer to detect whether photons are ↔ or ↕.

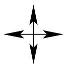
We must be more general. We must introduce an operator for each *observable*. For example, the operator for "+" polarization could be

$$\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$

Another operator, for "×" polarization, could be

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$$

(To avoid drawing the pictures



Standard Polarization

Hadamard Polarization

I'll use "+" and "×" respectively.)

The measurable outcomes correspond to the eigenvalues of the operator. Since these must be real numbers, the operator must be Hermitian. Thus the eigenvectors are orthogonal and, if normalized, provide a basis for the space. Thus the eigenvalues of $\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$ are $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$:

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

corresponding the the 45° polarization.

Suppose the photon to be measured were in the state

$$\mid \psi >= \begin{pmatrix} c \\ s \end{pmatrix}$$

and we are testing the observable (L for "look")

$$L = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \qquad \mid + >= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mid - >= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

L can detect only two states, $\mid + >$, to which it attributes value 1, and $\mid - >$, for which it sees value -1.

The probability that the original state $\mid \psi >$ goes either way, upon measurement, is the square of the corresponding amplitude, found by taking the inner product.

$$\begin{aligned} < \psi \mid + > &= (c, s)\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(c+s) \\ < \psi \mid - > &= (c, s)\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(c-s) \end{aligned}$$

We've extended our notation: the "bra", $< \psi \mid$, stands for the row vector for the state ψ; the "ket", $\mid + >$, stands for the column vector for the state $+$. Together they make a "braket", which is always, simply, a (possibly complex) number.

Thus the probabilities are

$$\begin{aligned} P_+ &= \frac{1}{2}(c+s)^2 = \frac{1}{2}(1+s_2) \\ P_- &= \frac{1}{2}(c-s)^2 = \frac{1}{2}(1-s_2) \end{aligned}$$

where $s_2 = 2cs$ (and as always $c^2 + s^2 = 1$).

These give, if we prepare a large number of identical states $\mid \psi >$, the proportion of the times measuring L yields $+1$ and the number of times it gives -1.

(Of course, if we keep *re*measuring the single state $\mid \psi >$, we will get an initial $+1$ or -1 with those probabilities, followed by repeats of the same value for every subsequent measurement, because the first measurment changes the state of the photon and

$$< + \mid + >= 1 =< - \mid - >$$

but

$$< - \mid + >= 0 =< + \mid - > .)$$

It is reasonable to ask about the *expected value* of the measurement of state $\mid \psi >$ by L. This is

$$< \psi \mid L \mid \psi >= (c, s)\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} c \\ s \end{pmatrix} = s_2$$

which also defines an extension of the bra-ket notation. To check, this should just be

$$P_+v_+ + P_-v_- = \frac{1}{2}(1+s_2)\times 1 + \frac{1}{2}(1-s_2)\times -1$$

where v_+ and v_- are just the eigenvalues—the respective possible outcomes of measurement.

Any hermitian operator can be expressed in terms of its eigenvalues and the projection onto its eigenvectors.

$$\begin{aligned}
L &= v_+ \mid + >< + \mid + v_- \mid - >< - \mid \\
&= 1 \times \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\frac{1}{\sqrt{2}}(1,1) + (-1)\times\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}\frac{1}{\sqrt{2}}(1,-1) \\
&= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}
\end{aligned}$$

We note that the bra-ket notation allows us to express a matrix—an operator—as a ket-bra.

This discussion tells us everything we can know about a given state $\mid \psi >$ in terms of the basis $\mid + >$ and $\mid - >$ provided by a given observable L. But it's going to be handy to represent this view of the state as a *density matrix*.

$$\begin{aligned}
\rho_{\psi L} &= P_+ \mid + >< + \mid \;+\; P_- \mid - >< - \mid \\
&= \frac{1}{2}(1+s_2)\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{2}(1-s_2)\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 1 & \\ & 1 \end{pmatrix} + \frac{1}{2}s_2\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}
\end{aligned}$$

Note that P_+ and P_- depend on ψ and $\mid + >$ and $\mid - >$, and that $\mid + >$ and $\mid - >$ depend on L.

We can get the expected value also from the density matrix and the observable.

$$\begin{aligned}
\mathrm{Tr}(\rho L) &= \mathrm{Tr}\left(\frac{1}{2}\begin{pmatrix} 1 & \\ & 1 \end{pmatrix} + \frac{1}{2}s_2\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\right) \\
&= \mathrm{Tr}\left(\frac{1}{2}\begin{pmatrix} & 1 \\ 1 & \end{pmatrix} + \frac{1}{2}s_2\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}\right) \\
&= s_2 \\
&= < \psi \mid L \mid \psi >
\end{aligned}$$

where the *trace*, Tr(), of a matrix is the sum of its diagonal elements.

Density matrices are hermitian because projection is hermitian. They are non-negative and have trace 1 because the projections are orthogonal and so a basis can be found in which each projection matrix has all zero entries except for a 1 at a unique point on the diagonal, and because the probabilities are non-negative and sum to 1.

In the special case that one of the probabilities is 1, the density matrix is its own square, because there is only one projection and projections are their own square.

Traces have come up twice in this discussion—$\mathrm{Tr}(\rho) = 1$ and $\mathrm{Tr}(\rho L)$. They are important because they do not depend on the coordinate system. The trace of a matrix is independent of the basis used to represent the matrix: trace is basis-invariant.

The trace of a density matrix is probability-weighted and is a form of averaging over the system it describes.

Density matrices become especially useful for composite systems only part of which can be measured. We can consider a system S and the environment E (we could call them Syl and Enn) and a state which is entangled between the two.

But first we look at the projection operators for S and E, both of whose states we suppose that we know.

$$\begin{array}{ccc} S & E & S\overset{\leftarrow}{\times}E \\ \begin{pmatrix} c \\ s \end{pmatrix} & \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} & \frac{1}{\sqrt{2}}\begin{pmatrix} c \\ c \\ s \\ s \end{pmatrix} \end{array}$$

Compare

$$\begin{aligned} \begin{pmatrix} c \\ s \end{pmatrix}(c,s)\overset{\leftarrow}{\times}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\frac{1}{\sqrt{2}}(1,1) &= \begin{pmatrix} c^2 & cs \\ cs & s^2 \end{pmatrix}\overset{\leftarrow}{\times}\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \\ \frac{1}{\sqrt{2}}\begin{pmatrix} c \\ c \\ s \\ s \end{pmatrix}\frac{1}{\sqrt{2}}(c,c,s,s) &= \frac{1}{2}\begin{pmatrix} c^2 & c^2 & cs & cs \\ c^2 & c^2 & cs & cs \\ cs & cs & s^2 & s^2 \\ cs & cs & s^2 & s^2 \end{pmatrix} \end{aligned}$$

All four of these expressions are equal. The first line is before taking the tensor product, the second line is after.

The traces of the two components before taking the tensor product are $c^2 + s^2$ and $1/2 + 1/2$ respectively. We can also find these two "partial" traces by looking at the 4-by-4 matrix (call it ρ).

For S

$$\begin{pmatrix} \rho_{00} + \rho_{11} & \rho_{02} + \rho_{13} \\ \rho_{20} + \rho_{31} & \rho_{22} + \rho_{33} \end{pmatrix}$$

For E

$$\begin{pmatrix} \rho_{00} + \rho_{22} & \rho_{01} + \rho_{23} \\ \rho_{10} + \rho_{32} & \rho_{11} + \rho_{33} \end{pmatrix}$$

In this special case, ρ is the density matrix for the whole system: only one projection is needed and its probability weight is 1.

The two 2-by-2 results are the *reduced density matrices*, ρ_S and ρ_E—and note that $\mathrm{Tr}(\rho_S) = 1 = \mathrm{Tr}(\rho_E)$ as well as $\mathrm{Tr}(\rho) = 1$.

Now let's look at the reduced density matrices for a system and environment entangled together. Suppose the overall state is $\mid \psi >= (\mid 01 > + \mid 10 >)/\sqrt{2}$.

$$\rho = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\frac{1}{\sqrt{2}}(0,1,1,0) = \frac{1}{2}\begin{pmatrix} 0 & & & \\ & 1 & 1 & \\ & 1 & 1 & \\ & & & 0 \end{pmatrix}$$

Since $\mid \psi >$ is not a product of separate states for S and for E (they are entangled) we cannot break it into two components and directly find ρ_S and ρ_E. But we can use the second method, extracting ρ_S and ρ_E from ρ.

$$\rho_S = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad\qquad \rho_E = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Note that these do not recombine, $\rho_S\overset{\leftarrow}{\times}\rho_E \neq \rho$,

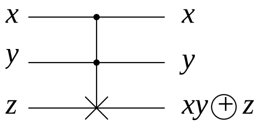
We note also that $\mid \psi >$ does not uniquely produce this result. It could be $(\mid 00 > + \mid 11 >)/\sqrt{2}$ or any of the Bell basis states,

So a measurement of observables in the system only, giving expected value $\mathrm{Tr}(\rho_S L_S)$ cannot tell us the exact state of system and environment combined. But it tells us everything it is possible for us to know about the *system.*

Ternary gates. We do not learn new principles by going to three qubits, but we fill in an important missing gate.

The only binary operator we have so far is **xor**. But we know from Boolean algebra that this does not give us a complete set of gates (see Note 4 of Week 10 and the related Excursion).

We will need at least an **and** operator. This can be provided by *two* control lines which are effectively **and**ed together.



x	y	z	$xy\oplus z$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

This is the controlled-controlled-not, **ccnot**, or *Toffoli* gate. It is also a reflection.

$$\left(\begin{array}{cc|cc|cc|cc}
1 & & & & & & & \\
 & 1 & & & & & & \\
\hline
 & & 1 & & & & & \\
 & & & 1 & & & & \\
\hline
 & & & & 1 & & & \\
 & & & & & 1 & & \\
\hline
 & & & & & & & 1 \\
 & & & & & & 1 &
\end{array}\right)$$

This Note gives the basics of quantum computing. The qubit brings to the classical bit the quantum concepts of superposition, of amplitude, of entanglement and of measurement. Now some algorithms.

In Notes 39 to 41 the quantum Fourier transform exploits superposition to achieve exponential speedup over the classical "fast" Fourier transform. Note 43 uses the linearity of superposition to prove the "no-cloning" theorem, which is needed in Note 42, along with measurement, to provide secure key distribution. Note 44 uses amplitudes to get a square-root speedup for unstructured database search. Note 45 uses entanglement and measurement for error detection and correction. In Note 46, entanglement shows that quantum physics really is nonlocal. And Note 47 looks at the basic mathematics of actually building a quantum computer.

39. Binary Fourier transform. What is the effect of the Discrete Fourier Transform (Week 9, Notes 1–3) on the *bits* of the function it is applied to? We use F for 0 and T for 1 to stress that we're discussing bits and to avoid possible confusion with 0s and 1s.

Here are 1-bit, 2-bit and 3-bit discrete Fourier transforms applied to the 2^n numbers $0, 1, \cdots, 2^n - 1$ with $n = 1, 2$ and 3. (For example, $n = 2$: $0 = \mathtt{FF}$, $1 = \mathtt{FT}$, $2 = \mathtt{TF}$, $3 = \mathtt{TT}$.)

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} \mathtt{F} \\ \mathtt{T} \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathtt{F}+\mathtt{T} \\ \mathtt{F}-\mathtt{T} \end{pmatrix}$$

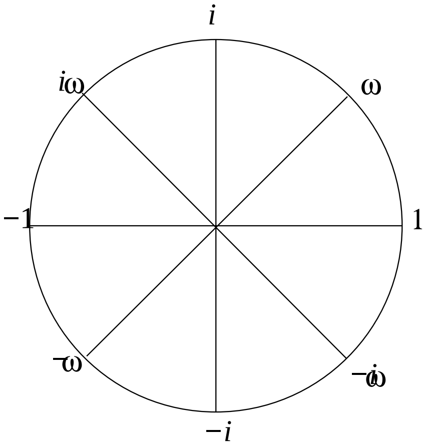
1 0

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} \mathtt{FF} \\ \mathtt{FT} \\ \mathtt{TF} \\ \mathtt{TT} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \mathtt{FF}+\mathtt{FT}+\mathtt{TF}+\mathtt{TT} \\ \mathtt{FF}+i\mathtt{FT}-\mathtt{TF}-i\mathtt{TT} \\ \mathtt{FF}-\mathtt{FT}+\mathtt{TF}-\mathtt{TT} \\ \mathtt{FF}-i\mathtt{FT}-\mathtt{TF}+i\mathtt{TT} \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} (\mathtt{F}+\mathtt{T})(\mathtt{F}+\mathtt{T}) \\ (\mathtt{F}-\mathtt{T})(\mathtt{F}+i\mathtt{T}) \\ (\mathtt{F}+\mathtt{T})(\mathtt{F}-\mathtt{T}) \\ \mathtt{F}-\mathtt{T})(\mathtt{F}-i\mathtt{T}) \end{pmatrix}$$

$$\frac{1}{\sqrt{2^3}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & i & i\omega & -1 & -\omega & -i & -i\omega \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & i\omega & -i & \omega & -1 & -i\omega & i & -\omega \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\omega & i & -i\omega & -1 & \omega & -i & i\omega \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & -i\omega & -i & -\omega & -1 & i\omega & i & \omega \end{pmatrix} \begin{array}{c} 2\,1\,0 \\ \begin{pmatrix} \mathtt{FFF} \\ \mathtt{FFT} \\ \mathtt{FTF} \\ \mathtt{FTT} \\ \mathtt{TFF} \\ \mathtt{TFT} \\ \mathtt{TTF} \\ \mathtt{TTT} \end{pmatrix} \end{array} = \frac{1}{\sqrt{2^3}} \begin{array}{c} 2 \qquad\quad 1 \qquad\quad 0 \\ \begin{pmatrix} (\mathtt{F}+\mathtt{T})(\mathtt{F}+\mathtt{T})(\mathtt{F}+\mathtt{T}) \\ (\mathtt{F}-\mathtt{T})(\mathtt{F}+i\mathtt{T})(\mathtt{F}+\omega\mathtt{T}) \\ (\mathtt{F}+\mathtt{T})(\mathtt{F}-\mathtt{T})(\mathtt{F}+i\mathtt{T}) \\ (\mathtt{F}-\mathtt{T})(\mathtt{F}-i\mathtt{T})(\mathtt{F}+i\omega\mathtt{T}) \\ (\mathtt{F}+\mathtt{T})(\mathtt{F}+\mathtt{T})(\mathtt{F}-\mathtt{T}) \\ (\mathtt{F}-\mathtt{T})(\mathtt{F}+i\mathtt{T})(\mathtt{F}-\omega\mathtt{T}) \\ (\mathtt{F}+\mathtt{T})(\mathtt{F}-\mathtt{T})(\mathtt{F}-i\mathtt{T}) \\ (\mathtt{F}-\mathtt{T})(\mathtt{F}-i\mathtt{T})(\mathtt{F}-i\omega\mathtt{T}) \end{pmatrix} \end{array}$$

In the $n = 3$ case, $\omega = (1+i)/\sqrt{2}$ is the 8th root of 1,



The pattern in this is that the 0 bit cycles through all 2^nth roots of 1 in the form $\mathtt{F} + (\sqrt[2^n]{1})\mathtt{T}$. The 1 bit cycles twice through the 2^{n-1}st roots of 1. And so on, with the period halving and the number of cycles doubling.

40. Quantum Fourier transform. In a quantum computer we can make each qubit simultaneously take on *all* possible values such as those we've computed above, and all the qubits together can simultaneously take on *all* the possible combinations above.

We just need quantum logic "gates" to do this to each qubit, controlled by its position, i.e., by the other qubits.

The $n = 1$ case tells us that we need the Hadamard gate

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

We'll see that the rest can be accomplished with H and a set of *phase change* gates

$$R_k = \begin{pmatrix} 1 & \\ & e^{2\pi i/2^k} \end{pmatrix}$$

$$R_0 = \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \qquad\qquad R_1 = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$
$$R_2 = \begin{pmatrix} 1 & \\ & i \end{pmatrix} \qquad\qquad R_3 = \begin{pmatrix} 1 & \\ & \omega \end{pmatrix}$$

But these will have to appear in *controlled* form so that the qubit positions will tell us which R_k to use and when to use it.

Since we have moved from bits, which can take on only the values `F` and `T`, to qubits, which can take on these values plus an infinity of intermediate values, we change notation to one using vectors.

$$\texttt{F} \text{ corresponds to } \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ written } | 0 >$$
$$\texttt{T} \text{ corresponds to } \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ written } | 1 >$$

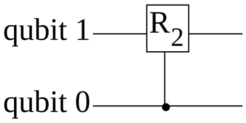
Applying the matrices representing H and R_2 gates to these vectors

$$\begin{aligned} | 0 > &\overset{H}{\rightarrow} \frac{1}{\sqrt{2}}(| 0 > + | 1 >) \\ | 1 > &\overset{H}{\rightarrow} \frac{1}{\sqrt{2}}(| 0 > - | 1 >) \\ | 0 > &\overset{R_2}{\rightarrow} | 0 > \\ | 1 > &\overset{R_2}{\rightarrow} i | 1 > \end{aligned} \qquad\qquad \begin{aligned} \begin{pmatrix} | 0 > \\ | 1 > \end{pmatrix} &\overset{H}{\rightarrow} \frac{1}{\sqrt{2}} \begin{pmatrix} | 0 > + | 1 > \\ | 0 > - | 1 > \end{pmatrix} \\ \begin{pmatrix} | 0 > \\ | 1 > \end{pmatrix} &\overset{R_2}{\rightarrow} \frac{1}{\sqrt{2}} \begin{pmatrix} | 0 > \\ i | 1 > \end{pmatrix} \end{aligned}$$

On the right above is a "compact" notation which should be taken only as a way of writing pairs of transformations together; we've seen in Note 38 that subtleties arise in different bases.

Using the notational equivalents (for bits) `F` $\leftrightarrow | 0 >$ and `T` $\leftrightarrow | 1 >$ we see that the 1-bit Fourier transform is just given by the Hadamard gate.

For more bits we'll need a *controlled* phase shift. Here qubit 0 controls the phase shift of qubit 1.

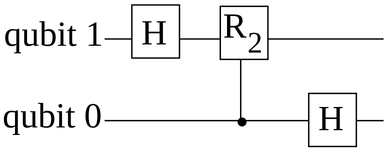


$$\begin{array}{c} 1\,0 \\ \begin{pmatrix} |00> \\ |01> \\ |10> \\ |11> \end{pmatrix} \end{array} \xrightarrow{\overset{0\ 1}{c}R_2} \begin{pmatrix} |00> \\ |01> \\ |10> \\ i\,|11> \end{pmatrix}$$

Only when the control qubit is `T` (1) does R_2 come into action. And then it affects only `T` (1) values of the controlled qubit.

(Recall that $| 00 >$ is just shorthand for $| 0 >| 0 >$, etc.)

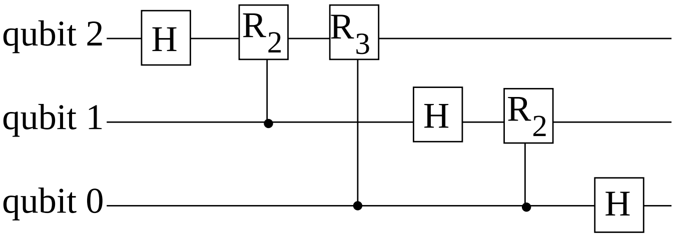
Let's try the following circuit for the 2-qubit Fourier transform.



$$\begin{array}{c} 1\,0 \\ \begin{pmatrix} |00> \\ |01> \\ |10> \\ |11> \end{pmatrix} \end{array} \xrightarrow{\overset{1}{H}} \frac{1}{\sqrt{2}} \begin{pmatrix} |00>+|10> \\ |01>+|11> \\ |00>-|10> \\ |01>-|11> \end{pmatrix} \xrightarrow{\overset{0\ 1}{c}R_2} \frac{1}{\sqrt{2}} \begin{pmatrix} |00>+|10> \\ |01>+i|11> \\ |00>-|10> \\ |01>-i|11> \end{pmatrix} \xrightarrow{\overset{0}{H}} \frac{1}{2} \begin{array}{c} 1 \qquad\qquad 0 \\ \begin{pmatrix} (|0>+|1>)(|0>+|1>) \\ (|0>+i|1>)(|0>-|1>) \\ (|0>-|1>)(|0>+|1>) \\ (|0>-i|1>)(|0>-|1>) \end{pmatrix} \end{array}$$

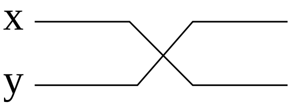
This is the same as the binary Fourier transform except that the qubits must be swapped at the end.

It is tedious but now straightforward to show that

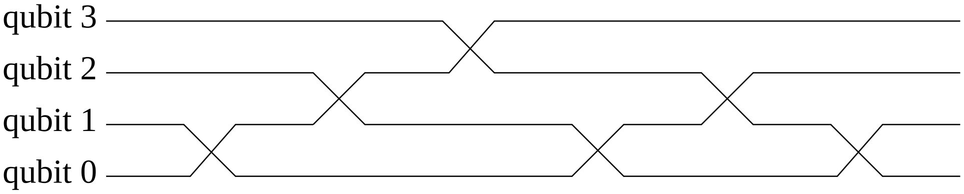


gives the same result as the 3-bit binary Fourier transform, except with the final bits in reverse order.

So we must learn how to swap bits. Let's introduce a swap gate.



We need $(n-1)n/2$ swap gates to reverse n bits.



That makes $3(n-1)n/2$ controlled-not gates if we follow Excursion *Implement swap gate.*

The cost of Fourier-transforming $N = 2^n$ qubits is $\mathcal{O}(n^2)$ quantum gates: n Hadamard gates, $(n-1)n/2$ phase-change gates and $3(n-1)n/2$ controlled-not gates ($(n-1)n/2$ swap gates).

That is $\mathcal{O}((\lg N)^2)$. The Fast Fourier Transform (FFT) of Note 5 in Week 9 is $\mathcal{O}(N \lg N)$. The Quantum Fourier Transform (QFT) is exponentially faster.

This speedup is due to the ability of the entangled quantum states involved to hold *all* possibilities—e.g., $| 00 > + | 01 > + | 10 > + | 11 >$ or $| 00 > - | 01 > +i | 10 > -i | 11 >$—simultaneously. The QFT exemplifies the exponential parallelism *sometimes* possible with quantum computing.

41. Finding periods. For the moment, the results of the quantum Fourier transform are still buried and inaccessible in the entangled state of the qubits. How do we use this QFT to find out, say, the period of a function?

Indeed, we haven't even applied the QFT to any function whose period we might wish to find.

Let's revisit the classical discrete Fourier transform (Week 9) to see how it finds periods. Suppose a 2-bit function is

x	0	1	2	3
$f(x)$	2	1	2	1

Then its Fourier transform is

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} 2 \\ 1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

where the 0-component, 3, of the result gives the mean value of $f()$—or would if we had normalized by $1/N$ instead of $1/\sqrt{N}$ ($N = 2^n = 4$ in this example). The other nonzero component gives the *period* of $f()$. Being the 2-component, it tells us $P = N/2 = 2$.

(This relationship between N, P and the indices of the nonzero Fourier results becomes clearer in an example with $N = 8$. Here is a function with period $P = 4$ and the result of the Fourier transform (the matrix is given in Note 39).

x	0	1	2	3	4	5	6	7
$f(x)$	2	1	3	0	2	1	3	0
$\sqrt{2}\times$FT	6	0	$-(1+i)$	0	4	0	$-(1+i)$	0

Note that the FT is nonzero for indices

$$\frac{N}{P} = 2, \qquad 2\frac{N}{P} = 4, \qquad 3\frac{N}{P} = 6$$

(and, of course, for index 0). So if we can find the indices of the nonzero results, the denominator gives the period. We needed something like this to crack secret codes in Excursion *Cracking RSA* in Week i—via Excursion *How many intervals?*.)

Now let's provide a function and see how the QFT can find its period.

A function$f()$ can be represented as a single quantum state by entangling its values: $\sum_{x=0}^{N-1} \mid x >\mid f(x) >$. For our earlier example

x	0	1	2	3
$f(x)$	2	1	2	1

this is, in decimal then in binary

$$\frac{1}{2}(\mid 0 >\mid 2 > + \mid 1 >\mid 1 > + \mid 2 >\mid 2 > + \mid 3 >\mid 1 >) =$$
$$\frac{1}{2}(\mid 00 >\mid 10 > + \mid 01 >\mid 01 > + \mid 10 >\mid 10 > + \mid 11 >\mid 01 >)$$

We could go further and replace, say, $\mid 00 >\mid 10 >$ by $\mid 0010 >$ but it is helpful to keep x and $f(x)$ visually separate. We'll talk about the "x-register" and the "f-register". (The 1/2 is the normalization factor.)

Now let's apply QFT to the x-register in this sum, so that, say, the first $\mid 00 >$ becomes (see Note 40)

$$\frac{1}{2}(\mid 00 > + \mid 01 > + \mid 10 > + \mid 11 >)$$

In the following I'm going to write these sums *vertically* for reasons that should become clear. The result of the QFT is

$$\begin{array}{rl}
\frac{1}{4}(& \mid 00 >\mid 10 > + \mid 00 >\mid 01 > + \mid 00 >\mid 10 > + \mid 00 >\mid 01 > \\
+ & \mid 01 >\mid 10 > +i \mid 01 >\mid 01 > - \mid 01 >\mid 10 > -i \mid 01 >\mid 01 > \\
+ & \mid 10 >\mid 10 > - \mid 10 >\mid 01 > + \mid 10 >\mid 10 > - \mid 10 >\mid 01 > \\
+ & \mid 11 >\mid 10 > -i \mid 11 >\mid 01 > - \mid 11 >\mid 10 > +i \mid 11 >\mid 01 >) \\
= \frac{1}{4}(& \mid 00 > (\mid 10 > + \mid 01 > + \mid 10 > + \mid 01 >) \\
+ & \mid 01 > (\mid 10 > +i \mid 01 > - \mid 10 > -i \mid 01 >) \\
+ & \mid 10 > (\mid 10 > - \mid 01 > + \mid 10 > - \mid 01 >) \\
+ & \mid 11 > (\mid 10 > -i \mid 01 > - \mid 10 > +i \mid 01 >))
\end{array}$$

$$
\begin{aligned}
= \tfrac{1}{2}(& \mid 00 > (\mid 10 > + \mid 01 >) \\
+ & \mid 01 > 0 \\
+ & \mid 10 > (\mid 10 > - \mid 01 >) \\
+ & \mid 11 > 0)
\end{aligned}
$$

Now if we measure the x-register we have zero amplitude for $\mid 01 >$ and for $\mid 11 >$ so we can only get 0 ($\mid 00 >$) or 2 ($\mid 10 >$) as the answer.

In the example with period 4 ($N = 8$) above, x would measure to be 0, 2, 4 or 6. The last three give, respectively,

$$\frac{N}{P} = 2 \qquad 2\frac{N}{P} = 4 \qquad 3\frac{N}{P} = 6$$

from which in each case

$$P = \frac{N}{2} = 4$$

But we do not know which of the multipliers 1, 2 or 3 applies when we've made just one measurement, so the best we can do to find P is to divide N by the result of the measurement.

The three possibilities are (still excluding 0—if we got 0 we'd have to run the Fourier transform again)

$$\frac{N}{2} = 4 \qquad \frac{N}{4} = 2 \qquad \frac{N}{6} = \frac{4}{3}$$

In two of these cases the numerator gives the period, 4. In the middle case we will mistakenly think that the period is 2.

We discussed this kind of reasoning in Week i, Excursion *How many intervals?* and what to do about it. (The full machinery of that Excursion is particularly germane if the period is not a power of 2.)

The upshot is that we have a reasonable chance of finding the period of a function from one measurement of its QFT.

This can be used to crack Rivest-Shamir-Adelman (RSA) encryption, which is the basis for almost all the encryption algorithms that keep our Internet transactions secure. RSA depends on the computational difficulty of factoring large integers, which can however be done by finding the period of a certain modular-arithmetic function—see Week i, Excursion *Cracking RSA*.

42. Quantum key distribution. Having cracked RSA can we replace it with a really secure system? RSA supports a public/private key pair with which anybody can encrypt (public key) a message which only one person can decrypt (private key). The security of RSA was provided by the apparent difficulty of factoring a large integer.

RSA is ultimately used to transmit a separate, one-time key which can be used to encrypt the actual message to be sent from, let's call her Fran, to, let's call him Tom. A random key, used only once, is guaranteed to be secure, as Claude Shannon has shown.

Such a "key distribution" can be done by quantum bits (qubits) directly, with security guaranteed by physics, not just by computational difficulty.

The idea proposed by Charles Bennett and Gilles Brassard (BB84) is to use two qubit encodings at random, transmit a number of qubits, then *subsequently* compare the *encoding methods*—but not the actual qubits.

Fran and Tom need a quantum channel, one-way from Fran to Tom, for the qubits, and a classical channel, two-way, subsequently to compare the encoding methods.

Let's suppose Fran wants to send Tom an $N = 4$ bit one-time key. They agree beforehand that each bit will be encoded, say as a polarized photon, randomly as either + or × (recall that + represents the standard basis and × represents the Hadamard basis and is obtained by the Hadamard transformation—a 22.5^o reflection or effectively a 45^o rotation). Fran's encoding sequence is not known to Tom so Tom must also randomly select one base or the other in sequence, to decode the photons.

To allow for incompatible coding steps, the number of qubits actually sent must double to $2N = 8$. And a further test for eavesdroppers, which we'll come to, will also expose N qubits, so the total must double again to $4N = 16$.

Here is an example.

Fran encodes	+	×	+	×	×	+	×	+	×	×	+	+	+	×	×	+
Fran sends	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	0
Tom decodes	+	×	×	+	×	×	+	×	+	×	×	+	+	+	×	×
Tom receives	0	1	?	?	1	?	?	?	?	0	?	0	1	?	0	?

On later comparison of the sequence of coding methods, both Fran and Tom discard the bits where the methods differed—those marked "?" in Tom's stream—whose results can agree only half the time because of the inherently probabilistic nature of quantum physics. So in this example they can use 7, or about $2N = 8$ qubits.

If there is an eavesdropper, both channels are vulnerable. We can do nothing to protect the classical channel but Fran and Tom can detect Eve eavesdropping on the quantum channel.

Eve knows the two coding methods but not their sequence, so she can do no better than to sequence them at random herself. Having absorbed Fran's photon at any step, Eve must send to Tom photons representing her own results using her same coding sequence.

Here is the same example, plus Eve.

Fran encodes	+	×	+	×	×	+	×	+	×	×	+	+	+	×	×	+
Fran sends	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	0
Eve decodes	+	×	+	×	+	×	+	×	+	×	+	×	+	×	+	×
Eve receives and sends	0	1	0	0	?	?	?	?	?	0	1	?	1	1	?	?
Tom decodes	+	×	×	+	×	×	+	×	+	×	×	+	+	+	×	×
Tom receives	0	1	?	?	$\textcircled{?}$	?	?	?	?	0	?	$\textcircled{?}$	1	?	$\textcircled{?}$	?

This time, when Tom and Fran compare coding sequences, there may be disagreement in two places (the circled question marks, for qubits 5, 12 and 15) about the values of the qubits. Of course they won't detect this yet because none of the qubits have been compared. But suppose they do compare some, say N, of the qubits associated with the agreeing coding steps, say bits 1, 5, 12 and 15 (the selection should be at random). For any bit Eve and Tom choose the same encoding methods with probability 1/2 in which case they get the same bit—which is what Fran sent. Or they choose different coding methods, in which case they will get the same bits half the time. So there is a 3/4 chance that Eve's snooping will not be noticed but a 1/4 chance that Eve will be caught. So Eve alters N/4 qubits on the whole, which is one qubit in this example. That could be any of the qubits 5, 12 or 15, And, of course, the more qubits the greater the likelihood of catching Eve.

If an eavesdropper has been detected, Fran and Tom can look for another channel and try again.

So BB84 gives us a probabilistically guaranteed secure communication algorithm which can be used to transmit one-time keys to encrypt messages (of the same length) over insecure classical channels.

43. No cloning. So why does Eve not just *copy* the qubits she receives, keeping one copy and passing the other on to Tom, then wait for Tom and Fran to compare coding methods over the (insecure) classical channel?

Because she can't. Unlike classical bits, qubits cannot, in general, be copied.

We would need a gate U which maps, say $\mid \psi 0 >$ to $\mid \psi\psi >$ thus making a copy of the first qubit,

$| \psi >$, in the second qubit which was originally zero.

The **cnot** gate does this if $| \psi >=| 0 >$ or if $| \psi >=| 1 >$

$$\begin{aligned} \textbf{cnot} :| 00 > &\rightarrow | 00 > \\ | 10 > &\rightarrow | 11 > \end{aligned}$$

but it does not copy an arbitrary qubit $| \psi >= \alpha | 0 > +\beta | 1 >$

$$\textbf{cnot} :| \psi 0 >\rightarrow \alpha | 00 > +\beta | 11 >$$

whereas

$$| \psi\psi >= (\alpha | 0 > +\beta | 1 >)(\alpha | 0 > +\beta | 1 >)$$

so these two are not the same, as they should be if **cnot** really mapped $| \psi 0 >$ to $| \psi\psi >$.

Wooters and Zurek (1982) and Dicks (1982) extended this argument to show that no gate will do. Suppose

$$U :| \psi 0 >\rightarrow| \psi\psi >$$

and that

$$| \psi >= \alpha | \phi > +\beta | \theta >$$

so, by linearity

$$U :| \psi 0 >\rightarrow \alpha | \phi\phi > +\beta | \theta\theta >$$

but, by hypothesis

$$U :| \psi 0 >\rightarrow (\alpha | \phi > +\beta | \theta >)(\alpha | \phi > +\beta | \theta >)$$

These two conclusions contradict each other.

44. Database search. Suppose we want to find the person with phone number `1111` in the telephone book

Name	Phone
`Ann`	`4444`
`Bob`	`5555`
`Cat`	`1111`
`Don`	`7777`
`Eve`	`3333`
`Flo`	`2222`
`Gio`	`8888`
`Hal`	`6666`

N	P
`A`	`4`
`B`	`5`
`C`	`1`
`D`	`7`
`E`	`3`
`F`	`2`
`G`	`8`
`H`	`6`

Name	Phone
`0001`	`0100`
`0010`	`0101`
`0011`	`0001`
`0100`	`0111`
`0101`	`0011`
`0110`	`0010`
`0111`	`1000`
`1000`	`0110`

(The middle table abbreviates the telephone book and the table on the right translates that to "nybbles" of ASCII: '`A`' is `41` in hexadecimal or the byte `01000001`; '`1`' is `31` in hexadecimal or `00110001`; and so on.)

If we had wanted to look up '`Cat`', for example, we would exploit the structure of the telephone book—its alphabetical order. For example a systematized version of what we do with a physical telephone book is a "binary search": start in the middle, see which way to go from there and iterate with each new half of the previous interval,

However, the telephone book is not structured for using *phone numbers* as the look-up key, so we'll essentially have to scan the whole book sequentially from beginning to end. Well, on the average over many searches, since we might succeed anywhere from entry number 1 to entry N (8 in this example), we must check $N/2$ entries. Compare this with lg$N$ for the binary search.

N	4	8	16	32	64	128	256	···	K	···	M	···	G
$N/2$	2	4	8	16	32	64	128	···					
lg$N$	2	3	4	5	6	7	8		10		20		30

For a quantum telephone book, however, this unstructured search is much less a problem. We can create a state which contains the result of *all* comparisons. For a Giga-entry file this would require 30 qubits.

For our first quantum telephone book let's take just four entries. We'll let $f()$ name the function we must represent. It has two values: 1 for a match and 0 for a non-match.

x	Name	Phone	f
00	Ann	4444	0
01	Bob	5555	0
10	Cat	1111	1
11	Don	7777	0

I've added an extra column x which is sinply the relative address, in computer memory, of each entry, so that $f(x)$ can be taken as a function of $x = 0, \cdots, 3$.

Now we do the usual thing and make a reversible gate using exclusive-or, $\oplus$.

$$U_f : \mid x > \mid q > \rightarrow \mid x > \mid q \oplus f(x) >$$

So for the two possible values of the single qubit q

x	q	$q \oplus f(x)$
00	0	0
00	1	1
01	0	0
01	1	1
10	0	**1**
10	1	**0**
11	0	0
11	1	1

Note that the result (apart from x, which just gets repeated) is the same as q where there is no match ($f(x) = 0$) but is flipped where $f(x) = 1$ (shown in bold). We can exploit this.

To see how to proceed we must think about the vectors given by the quantum amplitude of the state describing the quantum telephone book.

This is a multidimensional vector ($N = 4$ dimensions in the case of our reduced telephone book). Fortunately, the problem is essentially two-dimensional: whether there is a match or not.

We'll need two basis vectors.

$$\mid \alpha >= \frac{1}{\sqrt{N-M}} \sum_x (1 - f(x)) \mid x >= \frac{1}{\sqrt{3}} (\mid 00 > + \mid 01 > + \mid 11 >)$$

for non-matches, and

$$\mid \beta >= \frac{1}{\sqrt{M}} \sum_x f(x) \mid x >= \mid 10 >$$

for match (M is the number of entries which match a general query, $M = 1$ in this example, leaving $N - M$ non-matching entries).

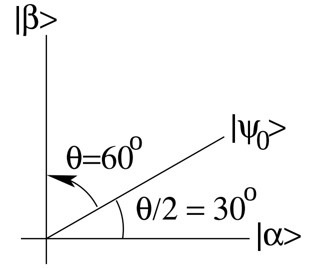
It is plausible to start our search with equal weights for all possibilities.

$$\begin{aligned} \mid \psi_0 > &= \frac{1}{\sqrt{N}} \sum_x \mid x > \\ &= \frac{1}{\sqrt{4}} (\mid 00 > + \mid 01 > + \mid 10 > + \mid 11 >) \end{aligned}$$

$$\begin{aligned} &= \cos\theta/2 \mid \alpha > + \sin\theta/2 \mid \beta > \\ &= \sqrt{\frac{N-M}{N}} \mid \alpha > + \sqrt{\frac{M}{N}} \mid \beta > \\ &= \sqrt{\frac{3}{4}} \mid \alpha > + \sqrt{\frac{1}{4}} \mid \beta > \end{aligned}$$

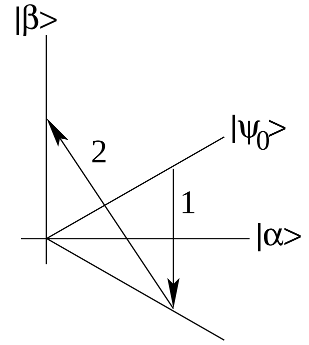
The middle line introduces the angle (called $\theta/2$ for reasons which will become clear next) that $\mid \psi >$ makes with $\mid \alpha >$ in the 2-dimensional space. In this example clearly $\theta/2 = 30$ degrees.

If we can rotate $\mid \psi_0 >$ by 60 degrees, it will wind up pointing exactly along $\mid \beta >$, which gives us the match we want.



Rotations are not quantum gate operations but reflections are, and two reflections make a rotation. To get the rotation by θ we make two steps:

- a) reflect $\mid \psi_0 >$ in $\mid \alpha$;
- b) reflect the result in $\mid \psi_0 >$.



Recall (from Note 19, Part IV) that a projection is the mean of the identity and the reflection

$$P = \frac{1}{2}(I + F)$$

so

$$F = 2P - I$$

And projection onto a vector $\vec{v}$ is given by the matrix that is the product of $\vec{v}$ with itself

$$P = \vec{v}\vec{v}^{\dagger}$$

For example, projection onto $\vec{v} = (c, s)^{\mathrm{T}}$ is

$$\begin{pmatrix} c \\ s \end{pmatrix} (c, s) = \begin{pmatrix} c^2 & cs \\ cs & s^2 \end{pmatrix}$$

Thus, reflection in the vector $(c, s)^{\mathrm{T}}$ is

$$\begin{aligned}
2\begin{pmatrix} c^2 & cs \\ cs & s^2 \end{pmatrix} - I &= 2\begin{pmatrix} c^2 & cs \\ cs & s^2 \end{pmatrix} - \begin{pmatrix} c^2+s^2 & \\ & c^2-s^2 \end{pmatrix} \\
&= \begin{pmatrix} c^2-s^2 & 2cs \\ 2cs & -(c^2-s^2) \end{pmatrix} \\
&= \begin{pmatrix} c_2 & s_2 \\ s_2 & -c_2 \end{pmatrix}
\end{aligned}$$

as we found in Note 19 (Part IV).

The notation we introduced in Note 38 extends to allow us to express both dot products of vectors

$$\vec{v} \cdot \vec{v} = \vec{v}^{\mathrm{T}} \vec{v}$$

and this new "outer product"

$$\vec{v}\vec{v}^{\mathrm{T}}$$

We've been writing $\mid v >$ for the column vector $\vec{v}$. Now we write $< v \mid$ for the corresponding row vector, $\vec{v}^{\mathrm{T}}$.

So

$$< v \mid v >= \vec{v}^{\mathrm{T}}\vec{v} = \vec{v} \cdot \vec{v}$$

is just a *number*, but

$$\mid v >< v \mid= \vec{v}\vec{v}^{\mathrm{T}}$$

is a matrix, or *operator*.

The first reflection we want, therefore, is

$$2 \mid \alpha >< \alpha \mid -I$$

since $\mid \alpha >< \alpha \mid$ is the projection onto vector $\mid \alpha >$.

To get this, we go back to the operator for the match function.

$$U_f =\mid x >\mid q >\rightarrow\mid x >\mid q \oplus f(x) >$$

We now exploit the flip we observed in the values of q when $f(x) = 1$. We define

$$\mid q_0 >= \frac{1}{\sqrt{2}}(\mid 0 > - \mid 1 >)$$

x	$\sqrt{2} \mid q_0 >$	$\sqrt{2} \mid q_0 \oplus f(x)$
00	$\mid 0 > - \mid 1 >$	$\mid 0 > - \mid 1 >$
01	$\mid 0 > - \mid 1 >$	$\mid 0 > - \mid 1 >$
10	$\mid 0 > - \mid 1 >$	$\mid 1 > - \mid 0 >$
11	$\mid 0 > - \mid 1 >$	$\mid 0 > - \mid 1 >$

This just changes the sign on $\mid q_0 >$, and hence on $\mid x >\mid q_0 >$, when there is a match. But that component of any vector is just the $\mid \beta >$ component, so the effect if U_f with $\mid q_0 >$ is just the reflection we wanted in the $\mid \alpha >$ axis.

We don't use the $2 \mid \alpha >< \alpha \mid$ operator.

(And since $\mid q_0 >$ is unchanged by U_f, except for that change of sign, it is often simply left out of the specification of U_f

$$U_f :\mid x >\rightarrow (-1)^{f(x)} \mid x >$$

.)

How do we get the second reflection

$$1 \mid \psi_0 >< \psi_0 \mid -I$$

out of implementable quantum gates?

A clue comes from the fact that

$$\mid \psi_0 >= W \mid 0 >$$

where W is the Walsh-Hadamard transformation of Excursion *Representing functions*:

$$W \mid 00 \cdots 0 >= H \otimes H \otimes \cdots H$$

So

$$W(2 \mid 0 >< 0 \mid -I)W = 2 \mid \psi_0 >< \psi_0 \mid -I$$

since W is its own inverse.

And that kernel just changes the sign of every amplitude except that of state $\mid 0 >$. It can be written as a π phase change for all but $x = 0$

$$2 \mid \psi_0 >< \psi_0 \mid -I = C_\pi$$

with

$$C_\pi \mid x >= (-1)^{\delta_{x0}-1} \mid x >$$

From all this we get the *Grover operator*

$$G = W C_\pi W U_f$$

(Lov Grover, 1996).

For the 4D case of our example we need only one application of the Grover operator to come up with the quantum state that gives the search result exactly. A measurement of $G \mid \psi_0 >$ then gives us 2 (from state $\mid 10 >$) with certainty.

For the 8D case

$$\sqrt{\frac{N-M}{N}} = \sqrt{\frac{7}{8}} = \cos\theta/2$$

and

$$\sqrt{\frac{M}{N}} = \sqrt{\frac{1}{8}} = \sin\theta/2$$

giving $\theta/2 = 20.7^{\text{o}}$ and $\theta = 41.48^{\text{o}}$.

Then the Grover operator brings us to $3\theta/2 = 62.1^{\text{o}}$ and a second rotation brings us to $5\theta/2 = 103.5^{\text{o}}$. Of these, 103.5^{o} is closer to 90^{o} and any further iteration will take us away from the match state. So we stop there and measure $GG \mid \psi_0 >$. The result is not guaranteed this time to give 2 because the amplitude of $\mid 10 >$ in the state is not 1 although it is close: 0.97.

In 16D three iterations bring us to 101.3^{o} and the match amplitude $\sin(101.3) = 0.98$ for 96% probabiity of finding the match.

In general, we take steps of size θ from $\theta/2$ to $\pi/2$ which will be

$$\text{round}\left(\frac{1}{\theta}\left(\frac{\pi}{2} - \frac{\theta}{2}\right)\right) = \text{round}\left(\frac{\pi}{2\theta} - \frac{1}{2}\right)$$

Since $\sin(\theta/2) = \sqrt{M/N}$ and, for $\theta/2 < 90^{\circ}$, $\sin(\theta/2) \le \theta/2$ we can say

$$\frac{\pi}{2\theta} = \frac{\pi}{4\text{invsin}(\sqrt{M/N}}) \le \frac{\pi}{4}\sqrt{\frac{N}{M}}$$

and so the number of iterations of the Grover operator is $\mathcal{O}(\sqrt{N})$

Thus the cost of Grover's algorithm for searching an unstructured database is much better than the classical $\mathcal{O}(N)$.

N	100	10000	mega
$N/2$	50	5000	500kilo
$\pi\sqrt{N}/4$	8	78	785

Note that the square root over the N in this complexity result is purely a quantum effect. It is this because the amplitude is given by square roots.

45. Detecting and correcting errors. Qubits are susceptible to errors in transmission and processing, just as are classical bits, and even more so. The no-cloning theorem of Note 43 seemed to present an obstacle to using redundancy—copy the bit, transmit all copies and compare them on reception—to detect and fix an error. But there are ways.

The **cnot** gate cannot duplicate an arbitrary state, but it can duplicate the *basis* vectors. In order to be able to test for a *majority* after transmission, we *triplicate* them.

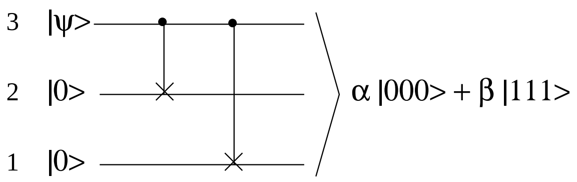
$$\overset{321}{|\psi 00>} = (\alpha|0> + \beta|1>)\,|0>\,|0>$$

$$= \alpha\,|0>|0>|0> + \beta\,|1>|0>|0>$$

$$\xrightarrow{\overset{3\ 2}{c\times}} \alpha\,|0>|0>|0> + \beta\,|1>|1>|0>$$

$$\xrightarrow{\overset{3\ 1}{c\times}} \alpha\,|0>|0>|0> + \beta\,|1>|1>|1>$$

$$= \alpha\,|000> + \beta\,|111>$$



This does not give us a product state of three copies of the original state $|\psi>$, but an entangled state of three qubits.

That's the setup. Next, the three qubits are transmitted and in transmission are subject to noise, $\mathcal{N}$. The kinds of error could be a bit-flip, $\mathcal{N} = X$, corresponding to a classical bit error. Or it could be a continuous phase rotation, $\mathcal{N} = e^{i\alpha X} = I\cos\alpha + iX\sin\alpha$ Or it could be a phase flip, $\mathcal{N} = Z$ (remember from Note 38 that X and Z are Pauli matrices). And so on. We will discuss these three kinds of error.

Note that the noise in each case operates on each transmitted qubit independently and, we assume, with sufficiently low probability p that $p^2 \approx 0$ (well, $p^2 \ll 1$) and so we can assume that only one of the three qubits has been affected, if at all.

Thus, a majority comparison will tell us not only if there has been a single-qubit but also which qubit and how to correct it.

We start with a bitflip

$$\mathcal{N}(\alpha \mid 000> + \beta \mid 111>) = \begin{cases} \alpha \mid 000> + \beta \mid 111> \\ \alpha \mid 001> + \beta \mid 110> \\ \alpha \mid 010> + \beta \mid 101> \\ \alpha \mid 100> + \beta \mid 011> \end{cases}$$

each alternative having probability $p(1-p)^2$. except the first, no-error, case, which has probability $(1-p)^3$. (I've left out the four other possibilities, involving probabilities $p^2(1-p)$ and p^3 because they are negligible by assumption.)

Note that the bitflip works on both terms: 0 → 1 and 1 → 0 for whichever qubit gets flipped.

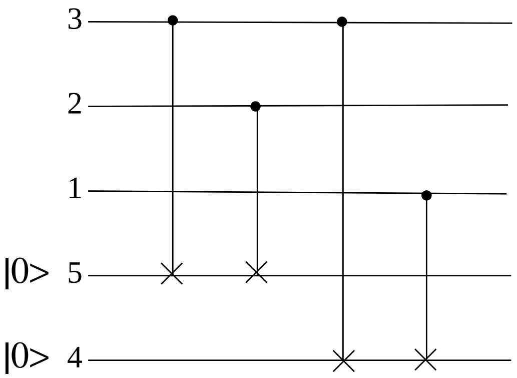
Now the receiver (I guess we're talking about Tom again, along with Fran, and Chas as the channel in case we must personify that too) must detect any error *without* disturbing the transmitted message. So Tom cannot just measure any of the incoming qubits because that would immediately disentangle the state, say

from	to	prob	or to	prob
$\alpha \mid 000 > +\beta \mid 111 >$	$\mid 000 >$	$\mid \alpha \mid^2$	$\mid 111 >$	$\mid \beta \mid^2$
$\alpha \mid 100 > +\beta \mid 011 >$	$\mid 100 >$	$\mid \alpha \mid^2$	$\mid 011 >$	$\mid \beta \mid^2$

etc.

What Tom must do is entangle them further, and break *that* entanglement by measuring. Tom uses four more **cnot** gates with two $\mid 0 >$ "ancilliary" qubits as targets. Let's say

$$(\alpha\,|\overset{321}{010}>+\beta\,|\overset{321}{101}>)|00>$$
$$=\alpha|\overset{32154}{01000}> + \beta|\overset{32154}{10100}>$$
$$\xrightarrow{\overset{3\,5}{{}^{c}X}} \alpha\,|01000>+\beta\,|10110>$$
$$\xrightarrow{\overset{2\,5}{{}^{c}X}} \alpha\,|01010>+\beta\,|10110>$$
$$\xrightarrow{\overset{3\,4}{{}^{c}X}} \alpha\,|01010>+\beta\,|10111>$$
$$\xrightarrow{\overset{1\,4}{{}^{c}X}} \alpha\,|01010>+\beta\,|10110>$$
$$=(\alpha\,|010>+\beta\,|101>)|10>$$



This restores the original state and gives a combined state which is the product of the original state and $\mid 10 >$, and hence disentangled. Furthermore, $\mid 10 >=\mid 2 >$, telling Tom directly that qubit 2 had been flipped. All Tom must do is measure the new ancilliary qubits 4 and 5 to find this out with probability 1 and without further disturbing the original state.

To establish that this sequence, $\overset{35}{{}^cX}$, $\overset{25}{{}^cX}$, $\overset{34}{{}^cX}$, $\overset{14}{{}^cX}$, applied to complementary bases $\mid x_3x_2x_1 >$ and $\mid x_3'x_2'x_1' >$ will a) give the same ancilliary results and b) identify the minority qubit, we need to know three things.

First, the sequence $\overset{35}{{}^cX}$, $\overset{25}{{}^cX}$ applied to $x_5 = 0$ yields $x_5 = x_3 \oplus x_2$ and the sequence $\overset{34}{{}^cX}$, $\overset{14}{{}^cX}$ applied to $x_4 = 0$ yields $x_4 = x_3 \oplus x_1$.

Second, $x \oplus y = x' \oplus y'$ simply tells us that x and y are different—true for both x,y and the complements x',y'.

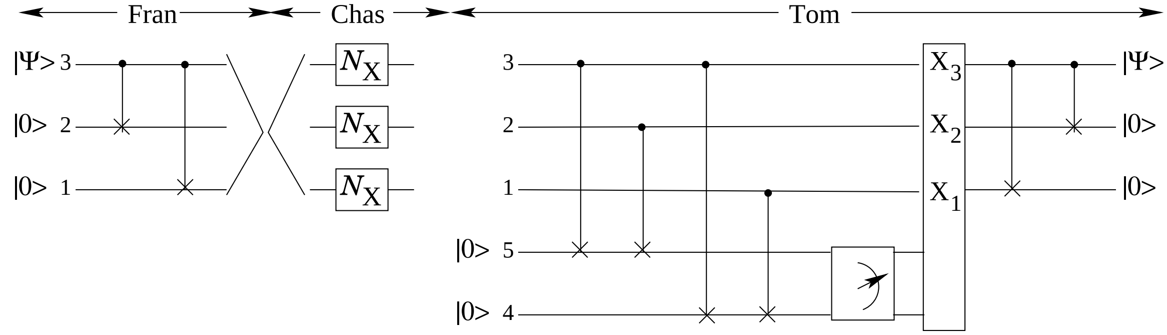
Third, if we rename $y = y_1y_0 = x_5x_4$ and $x = x_3x_2x_1$, then $y_1 = x_3 \oplus x_2$ tells us x_3 and x_2 differ, while $y_0 = x_3 \oplus x_1$ tells us x_3 and x_1 differ. So if x_3 is the minority qubit, $y = y_1y_0 = 11 = 3$; if x_2

is the minority qubit, $y = y_1 y_0 = 10 = 2$; and so on down to $y = 0$ if there is no minority qubit.

After Tom measures qubits 4 and 5 the answer tells him which qubit to correct, which he does with the appropriate **not**, $X_y, y = 1, 2, 3$; if $y = 0$ no correction is needed.

Finally it is polite—indeed dangerous not—to disentangle the three transmitted bits. This can be done by repeating Fran's original **cnot**s since they are self-inverses,

Here is the whole error-correction process schematically.



What if the channel noise were not a classical bitflip 0 ↔ 1 but some quantum superposition of single bitflips? Tom's "error syndrome" will now be a linear combination

$$\begin{array}{lll}
\alpha \mid 000 > +\beta \mid 111 > \overset{\text{error}}{\rightarrow} & c_0(\alpha \mid 000 > +\beta \mid 111 >) \quad \overset{\text{check}}{\rightarrow} & c_0(\alpha \mid 000 > +\beta \mid 111 >) \mid 00 > \\
 & c_1(\alpha \mid 001 > +\beta \mid 110 >) & c_1(\alpha \mid 001 > +\beta \mid 110 >) \mid 01 > \\
 & c_2(\alpha \mid 010 > +\beta \mid 101 >) & c_2(\alpha \mid 010 > +\beta \mid 101 >) \mid 10 > \\
 & c_3(\alpha \mid 100 > +\beta \mid 011 >) & c_3(\alpha \mid 100 > +\beta \mid 011 >) \mid 11 >
\end{array}$$

where $\mid c_0 \mid^2 + \mid c_1 \mid^2 + \mid c_2 \mid^2 + \mid c_3 \mid^2 = 1$.

Tom's measurement of the last two qubits will collapse this linear combination into one of the four components. That result gives unambiguous corrections—either X_3 if the measurement picks out the $(\alpha \mid 100 > +\beta \mid 011 >) \mid 11 >$ state, or X_2 or X_1 or no correction at all. In any case the result will be Fran's original qubit because the $\mid x >$ component of the state is selected along with the $\mid y >$ component.

The phase flip error,

$$\mathcal{N}_Z : \frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >) \leftrightarrow \frac{1}{\sqrt{2}}(\mid 0 > - \mid 1 >)$$

is just a bitflip error $\mid + > \leftrightarrow \mid - >$ in the Hadamard basis. So the bitflip circuit can be modified easily to a phase flip circuit by putting Hadamard gates on each bit immediately before and immediately after the channel.

Any linear combination of phase flip errors can also be corrected by this modified circuit.

Peter Shor in 1995 combined bitflip and phase-flip error corrections in a circuit which transmits nine qubits for each original qubit. He collaborated with others to reduce that requirement to seven then to five qubits. This involved formulating for qubits classical error-correcting codes such as Hamming's. Let's start with parity bits.

A way to detect a single-bit error in a code of n bits is to append an $n + 1$st bit and insist that the total number of 1-bits be even. Thus for $n = 3$ the value 2 would be encoded 0101 where the left three bits $d_2 d_1 d_0$ are data, and the rightmost bit p is the *parity bit*, set to 1 to ensure an even number of 1-bits. On the other hand, the value 5 would be coded 1010.

If any of the bits is flipped during transmission, including the parity bit, the count will no longer be even and we will know there has been an error.

However, unlike the case of triple redundancy, we don't know *where* the error is.

In 1950 Richard Hamming noticed that we can employ multiple parity bits, $A, B, C, \cdots$, to detect *and* locate errors if we name the data bits $AB, AC, BC, \cdots, ABC, \cdots$. Let's see how it works with three parity bits and four data bits.

data				parity		
AB	AC	BC	ABC	A	B	C
1	0	1	1	0	1	0

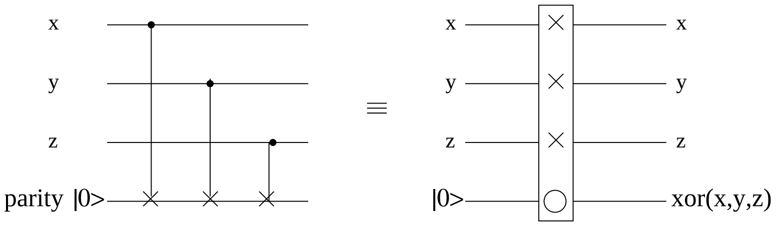
The idea is that the number of 1-bits with A in their name must be even, the number of 1-bits with B in their name must be even, and so on for each letter.

Now if there is a bitflip in data bit AB, re-running the parity check will show that parity bits A and B disagree with the received values. This points us straight to the data bit AB. And so on for each of the four data bits.

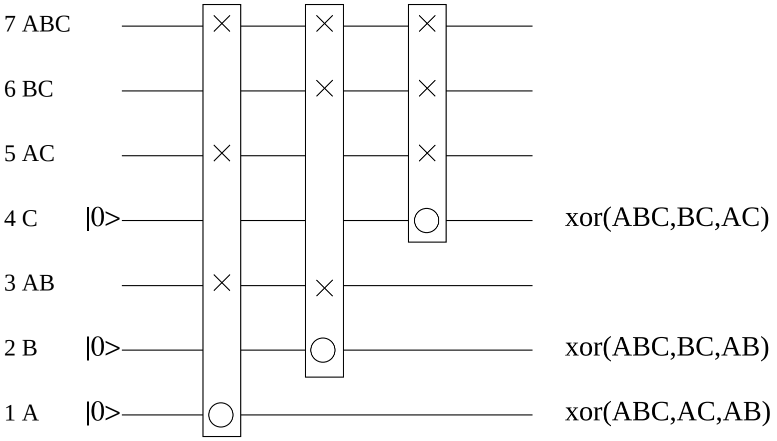
Note that if a data bit has been flipped then at least two parity bits disagree. If a parity bit has been flipped in transmission then that parity bit alone disagrees.

Implementing Hamming code classically is straightforward because any parity bit is simply the **xor** of all the bits it is checking. And since **xor** is fundamental to quantum computing this extends readily to qubits.

It is worth introducing a special **xor** gate which combines controlled-nots. For our 3-bit example we have



Then Hamming's 7-bit encoding with 3 parity qubits just requires us to number the qubits carefully.



That is at Fran's end. When the qubits are transmitted to Tom, he uses three auxiliary qubits, initially $| 0 >$, to recalculate the parity qubits, then measures the differences (**xor** again) between each one and the corresponding transmitted parity qubit. The resulting number, cba with $a = A \,\mathbf{xor}\, A'$, etc., identifies the qubit, if any, flipped by the transmission.

As well as transmission errors, of course, there are processing errors. *Fault-tolerant* processors use redundancy and error-correcting codes to catch faults at each qubit for each gate before they propagate to other gates. A robust process must tolerate faults in preparing initial states, gate processing, measurement, and even error correction.

46. Nonlocality: Einstein-Podolsky-Rosen. If we take two entangled particles such as the pair we introduced in Note 38, in state

$$\frac{1}{\sqrt{2}}(\mid 01 > + \mid 10 >)$$

and separate them far enough, we encounter the nonlocality of quantum physics.

For if we measure the first particle, say at position X, the state will collapse to $\mid 01 >$ or to $\mid 10 >$. This determines the value of the second particle, say at position Y. A measurement at position Y, even if far enough from X that a light signal could not connect the two measurements, will show the second particle as complementary to the first.

This does not imply faster-than-light communication, however. The results at both X and Y will appear random. Only if those results are later brought to a common place and compared will it become clear that the sequences of measurements are always complementary.

Maybe it's not all that bad. If I mailed a pair of gloves, one glove to Melbourne and the other glove to London, and if I instructed the recipients to open their packages at one fixed time (say, midnight and noon on a certain day in their respective time zones), then one would find a left glove and know that the other has a right glove, or vice-versa. But nobody would suppose that opening a package in any way influenced the other package.

Well, it *is* that bad, because polarization (or spin) states are not gloves. Each measurement can be made in a multitude of ways. not just left versus right, but spin in any one of an indefinite number of directions. And finding "spin-up" in a given direction at X immediately requires "spin-down" in the *same direction* at Y.

Albert Einstein, Boris Podolsky and Nathan Rosen in 1935 considered this "spooky action at a distance"—or nonlocality—to offend common sense to the point of establishing that quantum physics is incomplete. (Schrödinger in the same year coined the term "entanglement" to describe the binding of the two particles.) A complete theory, they said, must credit the particles with advanced knowledge of how they would respond to measurements, via what are called "hidden variables".

To look at this a little more closely it may be clearer to discuss an entangled state with both particles behaving the *same*.

$$\frac{1}{\sqrt{2}}(\mid 00 > + \mid 11 >)$$

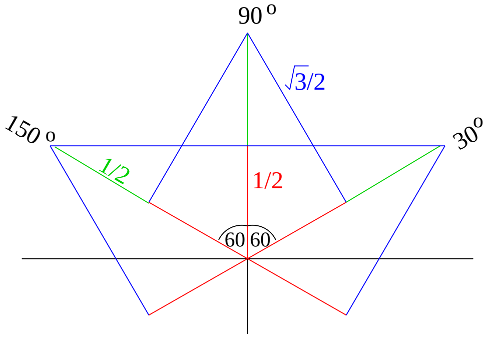
We can talk in terms of photons and polarization.

If we have a polarizer at X whose direction can be randomly set in any one of three different directions, and ditto at Y, what does quantum physics say will happen?

There are nine possible combinations. We look at probabilities—not of measurements being made in any particular direction, but just of the two measurements being the *same*, namely both photons get through the respective polarizers or both are blocked. If the polarizer directions are 30°, 90° and 150° at both X and Y positions, here are the probabilities for a match in each case.

$X\backslash Y$	30°	90°	150°
30°	1	1/4	1/4
90°	1/4	1	1/4
150°	1/4	1/4	1

Here is where those probabilities come from.



The projection of each direction on each other direction is $1/2$ (positive or negative doesn't matter: the polarization doesn't distinguish → from ←). These are the amplitudes, whose squares are the probabilities.

Thus if the polarization measured at X collapses the state in the 30° direction then the polarization at Y will also be in the 30° direction. But if the measurement at Y were made with the 150° polarizer, we see only the amplitude, $1/2$, of the 30° direction in the 150° polarizer.

This diagram shows all the possibilities for both photons *passing* their respective polarizers. The diagram for *absorption* is just rotated 90 degrees and gives the same numbers.

So the overall probability of both photons passing, or both photons being absorbed by, their respective polarizers, is $1/9$ of the sum of these

$$\frac{1}{9}\left(3+\frac{6}{4}\right)=\frac{1}{2}$$

John Stewart Bell in 1964 worked out the corresponding probabilities supposing that there *are* hidden variables. These could work any way we can imagine or even ways we cannot imagine, but in the end they record a predisposition for the two photons to pass (`P` for pass) or be absorbed (`A` for absorb) by each of the three polarizers. There are eight possible predispositions a photon can have for the 30°, 90° and 150° directions, respectively: `PPP, PPA, PAP, PAA, APP, APA, AAP` and `AAA`.

These can be related to the nine possible measurements. A check means that the predispositions of that row gives matching measurements, i.e., `P` and `P` or `A` and `A`, for that column.

	30,30	30,90	30,150	90,30	90,90	90,150	150,30	150,90	150,150
PPP	√	√	√	√	√	√	√	√	√
PPA	√	√		√	√				√
PAP	√		√		√		√		√
PAA	√				√	√		√	√
APP	√				√	√		√	√
APA	√		√		√		√		√
PPA	√	√		√	√				√
AAA	√	√	√	√	√	√	√	√	√

Apart from the `PPP` and `AAA` rows, which give probability 1 for a match, all the other rows give probability $5/9$ for a match.

Thus *any* hidden-variable theory predicts that the probability for a match must be $> 5/9$. This contradicts the result from quantum physics, and can be tested by experiment.

All the experiments agree with quantum physics.

Quantum physics then *does* invoke spooky action at a distance. It is *nonlocal.*

47. Building a quantum computer. How to do this is a discussion involving more physics than

we are prepared for. But we can say one important thing in general.

Our quantum gates so far have been reflections.

$$\begin{array}{ccc} \text{not} & \text{Hadamard} & \text{cnot} \\ \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} & \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{pmatrix} \end{array}$$

But the physical operations needed by an implementation are, by and large, rotations. For example a quantum computer based in NMR (nuclear magnetic resonance) uses radio-frequency pulses to rotate nuclear spins. A quantum computer based on cooled (almost motionless) ions trapped by electromagnetic fields use laser pulses to alter the excitations of the ions—still waves with rotating phases. Any two-state quantum system—which is what we need to represent qubits—is mathematically equivalent to a spin with operations

$$\vec{S} = (S_x, S_y, S_z) \quad \text{with} \quad S_j = \frac{1}{2}\hbar\sigma_j$$

using the Pauli matrices

$$\sigma_x = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \qquad \sigma_y = \begin{pmatrix} & -i \\ i & \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$

Note that the σ_j are reflections, but the $\sigma_j/2$ are, as we saw in Note 21, the generators for rotations.

So we'd like to be able to express quantum gates in terms of the half-Pauli matrices. The generator of a rotation through an angle ϕ is

$$e^{-1\phi\sigma/2}$$

and the physics is that a pulse of angular frequency ω and duration τ (and so a phase change $\phi = \omega\tau$) requires energy

$$\omega S = \omega\frac{\sigma}{2}\hbar$$

So we will need to express the gates in terms of these pulses.

As long as $\sigma^2 = I$, which is true for the Pauli matrices, any

$$e^{-i\alpha\sigma} = I\cos\alpha - i\sigma\sin\alpha$$

and this is an important relation we can exploit.

The **not** gate is easy, since

$$\textbf{not} = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$$

We'll try

$$\begin{aligned} e^{-i\pi S_x/\hbar} &= e^{-i(\pi/2)\sigma_x} \\ &= I\cos -\frac{\pi}{2} + i\sigma_x \sin -\frac{\pi}{2} \\ &= -i\sigma_x \\ &= -i\begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \end{aligned}$$

Apart from the phase factor $-i = e^{-i\pi/2}$, this is the **not** gate. An *overall* phase factor does not matter in quantum physics, so the 180-degree rotation (in $e^{-i\pi\cdots}$) flips the qubit (**not**), and $\pi = \omega\tau$, given the frequency ω, tells us how long, τ, the pulse should be to flip it.

The Hadamard gate requires pulses in the other directions.

$$\begin{aligned}
e^{-i\pi S_z/\hbar} e^{-i(\pi/2)S_y/\hbar} &= e^{-i(\pi/2)\sigma_z} e^{-i(\pi/4)\sigma_y} \\
&= (I\cos\frac{\pi}{2} - i\sigma_z \sin\frac{\pi}{2})(I\cos\frac{\pi}{4} - i\sigma_y\sin\frac{\pi}{4}) \\
&= -i\sigma_z \frac{1}{\sqrt{2}}(I - i\sigma_y) \\
&= i\begin{pmatrix} -1 & \\ & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \\
&= -i\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
\end{aligned}$$

which is the Hadamard gate, also with an overall phase of $-i = e^{-i\pi/2}$.

By the way, the physical setup may support S_x and S_y but not S_z. We can get around this:

$$\begin{aligned}
e^{-i\phi S_z/\hbar} &= e^{-i(\pi/2)S_x/\hbar} e^{-i\phi S_y/\hbar} e^{i(\pi/2)S_x/\hbar} \\
I\cos\frac{\phi}{2} + i\sigma_z \sin\frac{\phi}{2} &= \frac{1}{\sqrt{2}} \begin{pmatrix} e^{-i\phi/2} & \\ & e^{i\phi/2} \end{pmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \\
&= (I\cos\frac{\pi}{4} - i\sigma_x \sin\frac{\pi}{4})(I\cos\frac{\phi}{2} - i\sigma_y\sin\frac{\phi}{2})(I\cos\frac{\pi}{4} + i\sigma_x\sin\frac{\pi}{4})
\end{aligned}$$

And you can also show

$$e^{-i\phi S_z/\hbar} = e^{-i(\pi/2)S_y/\hbar} e^{-i\phi S_z/\hbar} e^{i(\pi/2)S_y/\hbar}$$

Going to two qubit gates such as **cnot** involves tensor products.

Here is the March 2000 proposal by Debbie Leung, Isaac Chuang, Fumiko Yamaguchi and Yoshihisa Yamamoto. (We can relax and use $\otimes$ instead of $\overset{\leftarrow}{\times}$ until matrices are involved.)

$$\begin{aligned}
&(\quad e^{-i(\pi/2)S_y\hbar} \quad e^{i(\pi/2)S_x\hbar}\ e^{i(\pi/2)S_y\hbar}) \otimes (e^{-i(\pi/2)S_x\hbar}\ e^{i(\pi/2)S_y\hbar}) \times (e^{-i\pi(S_z\otimes S_z)\hbar^2}) \times (I \otimes e^{-i(\pi/2)S_y\hbar}) \\
= \quad & \frac{1}{\sqrt{2^7}}((I - i\sigma_y)(I + i\sigma_x)(I + i\sigma_y)) \otimes ((I - i\sigma_x)(I + i\sigma_y)) \times (I - i\sigma_z \otimes \sigma_z) \times (I \otimes (I - i\sigma_y)) \\
= \quad & \frac{1}{\sqrt{2^7}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \overset{\leftarrow}{\times} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \\
\times \quad & \begin{pmatrix} 1-i & & & \\ & 1+i & & \\ & & 1+i & \\ & & & 1-i \end{pmatrix} \left(I \overset{\leftarrow}{\times} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \right) \\
= \quad & \frac{1}{\sqrt{2^3}} \begin{pmatrix} 1+i & & & \\ & 1-i & & \\ & & 1+i & \\ & & & 1+i \end{pmatrix} \begin{pmatrix} 1+i & 1-i & & \\ -(1-i) & 1+i & & \\ & & 1+i & 1-i \\ & & -(1+i) & 1-i \end{pmatrix} \\
& \begin{pmatrix} 1-i & & & \\ & 1+i & & \\ & & 1+i & \\ & & & 1-i \end{pmatrix} \begin{pmatrix} 1 & -1 & & \\ 1 & 1 & & \\ & & 1 & -1 \\ & & 1 & 1 \end{pmatrix}
\end{aligned}$$

$$
\begin{aligned}
&= \quad \frac{1}{\sqrt{2}}\left(\begin{array}{cc|cc} 1-i & & & \\ & 1-i & & \\ \hline & & & 1-i \\ & & 1-i & \end{array}\right) \\
&= \quad \frac{1-i}{\sqrt{2}}\textbf{cnot}
\end{aligned}
$$

Note the overall phase factor of $e^{-i\pi/4}$.

II. **The Excursions**

You've seen lots of ideas. Now *do* something with them!

1. Show that **xor** is commutative. Show by induction on the number of bits that **xor** applied repeatedly just tells if the number of `T` bits is odd. Hence show that **xor** is associative.

2. Show that $H \otimes H$ applied to **cnot** (see Note 38) is

$$
\begin{aligned}
&\tfrac{1}{4}\left(\begin{array}{cc|cc} 1 & & 1 & \\ & 1 & & 1 \\ \hline 1 & & -1 & \\ & 1 & & -1 \end{array}\right)\left(\begin{array}{cc|cc} 1 & 1 & & \\ 1 & -1 & & \\ \hline & & 1 & 1 \\ & & 1 & -1 \end{array}\right)\left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)\left(\begin{array}{cc|cc} 1 & 1 & & \\ 1 & -1 & & \\ \hline & & 1 & 1 \\ & & 1 & -1 \end{array}\right)\left(\begin{array}{cc|cc} 1 & & 1 & \\ & 1 & & 1 \\ \hline 1 & & -1 & \\ & 1 & & -1 \end{array}\right) \\
&= \left(\begin{array}{cc|cc} 1 & & & \\ & & & 1 \\ \hline & & 1 & \\ & 1 & & \end{array}\right)
\end{aligned}
$$

3. How does the discussion in Note 38 of the numbers of entangled versus pure states change when we consider that the parameters $(\alpha, \beta, \cdots a, b, \cdots)$ are complex numbers?

4. a) Show that the entangled *Bell states* (see Note 38)

$$
\begin{aligned}
B_0 &= \frac{1}{\sqrt{2}}(|\ 00 > + \ |\ 11 >) \\
B_1 &= \frac{1}{\sqrt{2}}(|\ 00 > - \ |\ 11 >) \\
B_2 &= \frac{1}{\sqrt{2}}(|\ 01 > + \ |\ 10 >) \\
B_3 &= \frac{1}{\sqrt{2}}(|\ 01 > - \ |\ 10 >)
\end{aligned}
$$

   form an orthogonal basis for the 4-dimensional 2-qubit space. Hint: represent each state as a tensor product of vectors,
   b) Show that the Hadamard transformation, acting on each of the two qubits, has the effect

$$
\begin{aligned}
H \otimes H : B_0 &\rightarrow B_0 \\
B_1 &\leftrightarrow B_2 \\
B_3 &\rightarrow -B_3
\end{aligned}
$$

   c) What operations disentangle the Bell states?

5. Check that

$$
L \mid + >= (v_+ \mid + >< + \mid + v_- \mid - >< - \mid) \mid + >= v_+ \mid + >
$$

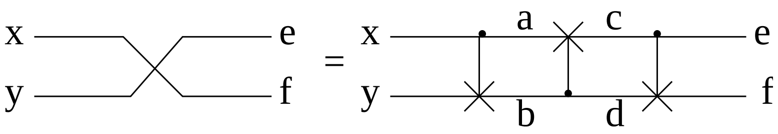
and

$$L \mid - >= (v_+ \mid + >< + \mid + v_- \mid - >< - \mid) \mid - >= v_- \mid - >$$

using only the rules for bras and kets given in Note 38, without expanding into matrices and vectors.

6. Note 38 presents quantum gates for **not** and **and**. From these we should be able to generate an **or** gate. The three, **and**. **or** and **not**, might help in designing at least classical boolean circuits from quantum gates.
   In analogy with the **and** gate the **or** gate should output $(x+y) \oplus z$ as well as the two inputs, in order to be reversible.
   Hint. Show that $(x \oplus y)' = x' \oplus y = x \oplus y'$.
   (I leave as an exercise the augmentation of this to a what-or-gate.)

7. **Implement swap gate.** Show that the swap gate can be implemented by three controlled-not gates.

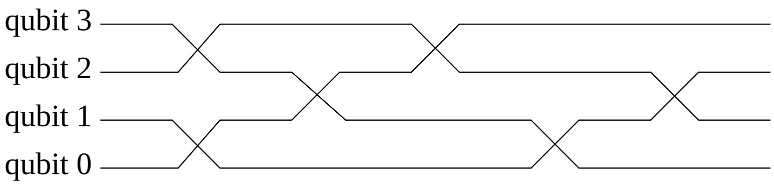


x	y	a	b	c	d	e	f
0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	0
1	0	1	1	0	1	0	1
1	1	1	0	1	0	1	1

In matrix terms

$$\begin{aligned}
\left(\begin{array}{cc|cc} 1 & & & \\ & & 1 & \\ \hline & 1 & & \\ & & & 1 \end{array}\right)\begin{pmatrix} 00 \\ 01 \\ 10 \\ 11 \end{pmatrix}
&= \left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)\left(\begin{array}{cc|cc} 1 & & & \\ & & & 1 \\ \hline & & 1 & \\ & 1 & & \end{array}\right)\left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)\begin{pmatrix} 00 \\ 01 \\ 10 \\ 11 \end{pmatrix} \\
&= \left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)\left(\begin{array}{cc|cc} 1 & & & \\ & & & 1 \\ \hline & & 1 & \\ & 1 & & \end{array}\right)\begin{pmatrix} 00 \\ 01 \\ 11 \\ 10 \end{pmatrix} \\
&= \left(\begin{array}{cc|cc} 1 & & & \\ & 1 & & \\ \hline & & & 1 \\ & & 1 & \end{array}\right)\begin{pmatrix} 00 \\ 10 \\ 11 \\ 01 \end{pmatrix} \\
&= \begin{pmatrix} 00 \\ 10 \\ 01 \\ 11 \end{pmatrix}
\end{aligned}$$

8. To reverse $N = 2^n$ bits (so N is a power of 2) show that divide-and-conquer does not change the number of swap gates. The idea would be to reverse two sets of $N/2$ bits then swap the results.



The induction step requires showing that

$$2\frac{(n/2-1)n/2}{2}+\frac{n}{2}\frac{n}{2}=\frac{(n-1)n}{2}$$

9. Check the $N = 8$ Fourier transform of Note 41 using the transformation matrix in Note 39.

10. **Representing functions.** a) Strictly speaking we should make the function representation in Note 41 *reversible*. That way we can apply the same transformation later to disentangle it. For a single value of $f()$ this would be given by the gate

$$U_f :\mid x, y > \rightarrow \mid x, y \oplus f(x) >$$

But in the discussion of Note 41 we can take $y = 0$ and the **xor** $0 \oplus f(x) = f(x)$
b) The representation of all values of x combined can be found by repeating the Hadamard transformation (Notes 38 and 40) for every bit. The result is called the Walsh, or Walsh-Hadamard, transformation

$$\begin{aligned} W \mid 00 > &= H \otimes H \mid 00 > \\ &= \frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >)\frac{1}{\sqrt{2}}(\mid 0 > + \mid 1 >) \\ &= \frac{1}{2}(\mid 00 > + \mid 01 > + \mid 10 > + \mid 11 >) \end{aligned}$$

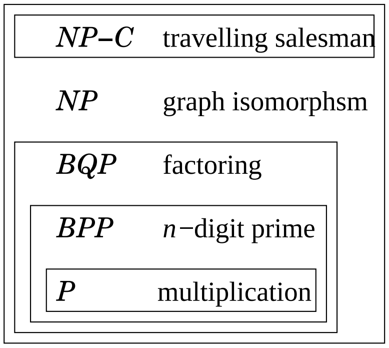
for 2 qubits, and in general for n qubits

$$W \mid 0 >= \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1} \mid x >$$

where we've slipped into multibit $|>$ symbols.
c) We can apply U_f to this (note $y = 0$)

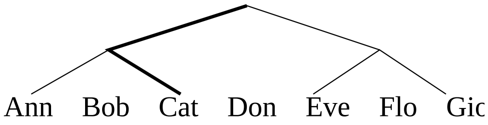
$$U_f\frac{1}{\sqrt{N}}\sum \mid x >\mid 0 >= \frac{1}{\sqrt{N}}\sum \mid x >\mid f(x) >$$

11. Look up the $\mathcal{BQP}$ class of "computational complexity". Simplistically, computational complexity distinguishes between algorithms which run in polynomial time (class $\mathcal{P}$) from those that run in exponential time. Algorithms in class $\mathcal{NP}$ appear to need exponential time to run but, once the answer is found, it can be *checked* in polynomial time.
Multiplication of an m-bit by an n-bit number is polynomial, needing $\mathcal{O}(mn)$ basic operations. So factoring an integer, which is exponential in time, falls into $\mathcal{NP}$ because it can be checked by multiplication.
If we allow probabilistic algorithms, such as the period-finding algorithm of Note 41, some intermediate complexity classes appear, including $\mathcal{BPP}$ (bounded error probabilistic polynomial) and $\mathcal{BQP}$ (bounded error quantum polynomial).
Here is a quick overview, showing class inclusions, with one example algorithm for each class.



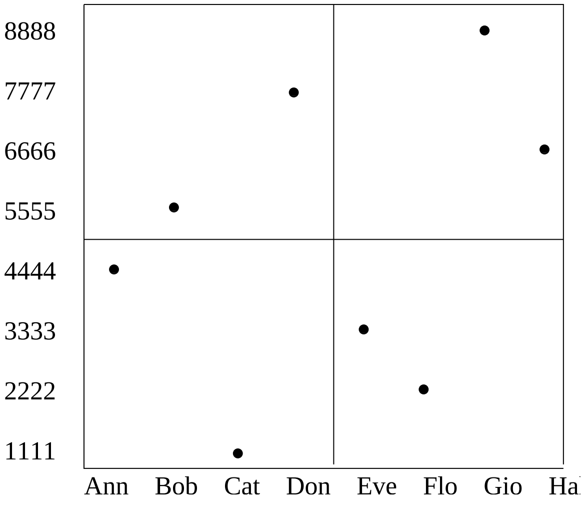
“$\mathcal{NP}-\mathcal{C}$” stands for NP-complete. You can look it up.

12. Here is a binary search on the telephone book of Note 44. Mersenne files (of sizes 2^n-1 entries) make the neatest illustration of binary search, so (shades of “2001: A Space Odessey”) we get rid of `Hal` for this example.



The graph is the “search tree” for all possible searches of this file. The heavy line shows the search for `Cat`: `Don`, in the middle, is too high alphabetically, so we branch left; `Bob`, in the middle of the left half, is too low, so we branch right; `Cat`, in the middle of the right half of that half, matches, so we stop successfully.
If there were 2^n-1 entries in the file, how many of them must we check in a binary search: in the worst case? on the average?
If you have an insulated wire with an electrical break somewhere inside it, and the only way you could test for electrical breaks is by attaching probes to the two ends of the wire, then, instead of throwing the whole wire out, you might cut it in various places to produce new ends to probe. How would you find the break after making the fewest cuts?

13. It is possible to structure a database so that classical techniques can operate in $\sqrt{N}$ time although that is not guaranteed as an upper bound.
Here is the telephone book of Note 44 stored as a *multidimensional paging* structure (Ekow Otoo, 1984). Note that it is simultaneously sorted on *both* fields.



The four pages shown are the retrieval units, as would be especially suited to *secondary storage* such as magnetic disc or flash memory. The data are distributed such that two entries are stored on each page. In the search for phone number `1111` only two pages need be retrieved ($2 = \sqrt{4}$). Of course, that is also true of the search for, say, `Don`: the data are structured *symmetrically*. They are still structured, and so as to permit retrieval on either field. This does not trump Grover's ability to search an unanticipated field for which no structure has been prepared in advance.

14. Show that the continuous phase rotation error of Note 45 is a special case of the superposition of bitflip errors.

15. Show that

$$\begin{aligned}\alpha \mid 000 > +\beta \mid 111 > &= \frac{1}{\sqrt{2^3}}((\alpha+\beta)(\mid +++> + \mid +--> + \mid -+-> + \mid --+>)\\ &+\frac{1}{\sqrt{2^3}}((\alpha-\beta)(\mid ++-> + \mid +-+> + \mid -++> + \mid --->)\end{aligned}$$

(See the discussion of phaseflip errors inNote 45.)

16. Show how to extend the Hamming code of Note 45 so that four parity bits check eleven data bits; five and twenty-six; $\cdots$

17. Show that the Hamming code with two parity bits is just the triple redundancy discussed in Note 45.

18. To make an experiment for entangled state $(\mid 01> + \mid 10>)/\sqrt{2}$ comparable to that for $(\mid 00> + \mid 11>)/\sqrt{2}$ in Note 46 we must check for measurements which produce *opposites*, not matches. What direction angles should we compare at both positions?

19. What does the nonlocality of the EPR effect (Note 46) say about the arguments leading to faze (or gauge) theory (e.g., Note 16 of Part III), and about quantum fields in general?

20. Different books on quantum computing have different strengths. Here are four, in order of publication date within each topic.

Note	topic	citation
38	entanglement	[RP11, p.77]
	cautions	[RP11, p.80]
	density matrix	[NO08, p.39], [RP11, p.267]
41	period-finding	[NO08, p.113]
42	BB84 key distr.	[NO08, p.62]
43	no cloning	[SS04, p.56], [NO08, p.75]
44	database srch.	[SS04, p.122]
45	error correct.	[NO08, p.196], [RP11, p.250]
46	EPR	[RP11, p.62]
47	NMR, ion gates	[SS04, p.158], [Meg08, p.361]

21. Any part of the Prefatory Notes that needs working through.

References

[Meg08] Zdzislaw Meglicki. *Quantum Computing without Magic: Devices.* The MIT Press, Cambridge MA, 2008.

[NO08] Mikio Nakahara and Tetsuo Ohmi. *Quantum Computing: From Linear Algebra to Physical Realizations.* CRC Press (Taylor & Francis), Boca Raton, 2008.

[RP11] Eleanor Rieffel and Wolfgang Polak. *Quantum Computing: A Gentle Introduction.* The M.I.T. Press, Cambridge MA, 2011.

[SS04] Joachim Stolze and Dieter Suter. *Quantum Computing: A Short Course from Theory to Experiment.* Wiley-VCH GmbH & Co. KGaA, Weinheim, 2004.