# $\hat{Q}\,|B\rangle$asics

**Activity 1:** Computing with matrices

## Problem 1: From truth tables to matrices

An XOR operation corresponds to adding two numbers mod 2 in binary, written with the $\oplus$ symbol. The truth table is below:

| $b_1, b_2$ | $b_1 \oplus b_2$ |
|---|---|
| 0,0 | 0 |
| 0,1 | 1 |
| 1,0 | 1 |
| 1,1 | 0 |

(a) Construct a matrix representation of XOR as a map from $\mathbb{R}^4 \to \mathbb{R}^2$

(b) Exhibit one way to make XOR invertible without adding extra bits. Write down the matrix and find its inverse.

*Note: The CNOT gate is just an XOR gate in disguise!*

### Solution

The general procedure for converting a truth table into a matrix is to first match elements in the domain and codomain with basis elements of a vector space. The canonical choice is to match binary numbers with standard basis vectors by their index. XOR maps $\{0,1\}^{\times 2} \to \{0,1\}$, so the domain is spanned by $(0,0) \leftrightarrow \vec{e}_0$, $(0,1) \leftrightarrow \vec{e}_1$, $(1,0) \leftrightarrow \vec{e}_2, (1,1) \leftrightarrow \vec{e}_3$. As a reminder, $\vec{e}_i$ denotes a vector of zeros with a 1 in the $i^{\text{th}}$ place. The codomain is spanned by $0 \leftrightarrow e_0$ and $1 \leftrightarrow e_1$. Our XOR is then represented by a linear map $M$ with $M\vec{e}_0 = M\vec{e}_3 = \vec{e}_0$ and $M\vec{e}_1 = M\vec{e}_2 = \vec{e}_1$, following the truth table above. Thus we construct a matrix representation of XOR by putting $\vec{e}_0$ in the $0^{\text{th}}$ and $3^{\text{rd}}$ columns and putting $\vec{e}_1$ in the $1^{\text{st}}$ and $2^{\text{nd}}$ columns:

$$M = \begin{pmatrix} \vec{e}_0 & \vec{e}_1 & \vec{e}_1 & \vec{e}_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

There are several possible ways to make XOR invertible by adding additional inputs. Without aditional bits, we can construct the map $XOR' : (a, b) \mapsto (a, a \oplus b)$. We can verify this map is invertible because $(a, a \oplus b \oplus a) = (a, b)$. The truth table for this operation is shown below. By identifying each binary number in the domain and codomain

| $b_1, b_2$ | $b_1,\ b_1 \oplus b_2$ |
|---|---|
| 0,0 | 0,0 |
| 0,1 | 0,1 |
| 1,0 | 1,1 |
| 1,1 | 1,0 |

with a standard basis vector, the matrix representation $M'$ is obtained:

$$M' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

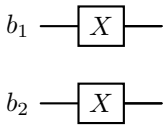
This is the same matrix as a CNOT gate which is conditioned on $b_2$ and acts on $b_1$. One can verify either through direct computation or by exploiting the block-diagonal form that this matrix is self-inverse.

# Problem 2: Building bigger circuits

Describing correlated bits is important for characterizing quantum circuits. Fortunately, simple operations on joint systems can often be described as block matrices, where each block corresponds to a single bit.

## Part a

Consider the following circuit:




Where $X$ is shorthand for the NOT gate. Write down the matrix form of this circuit. Can you express it in block form in terms of NOT matrices?

## Solution

Products of single-bit operators bear the same Kronecker-product structure as products of single-bit states. To see this, we first write out the truth table for $X_2X_1$. Following the procudure for identifying this binary map with a

| $b_1, b_2$ | $\bar{b}_1, \bar{b}_2$ |
|---|---|
| 0,0 | 1,1 |
| 0,1 | 1,0 |
| 1,0 | 0,1 |
| 1,1 | 0,0 |

linear operator, we find the matrix $M$ representing $X_2X_1$.

$$M = \begin{pmatrix} \vec{e}_3 & \vec{e}_2 & \vec{e}_1 & \vec{e}_0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Since $X$ interchanges the two basis vectors, the matrix representing $X$ is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. With this $M$ takes a block form:

$$M = \begin{pmatrix} 0 & X \\ X & 0 \end{pmatrix}$$

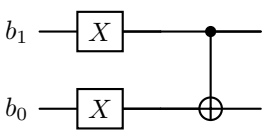
This is exactly like the Kronecker product of two states, where we have made a block matrix by taking the representation of $X$ and multiplying each of the matrix elements by the matrix $X$. In fact, just like with states, this is also referred to as the Kronecker product, and is denoted $X \otimes X$.

## Part b

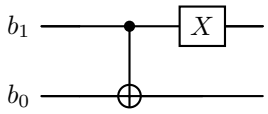
Now using the block-matrix form of the CNOT gate

$$\text{CNOT} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}$$

Can you write out the matrix of this circuit?



Use the above to prove that the previous circuit is equivalent to the following circuit




### Solution

Multiplying block matrices is exactly like multiplying matrices with scalar entries, but order can matter because some matrices may not commute. The circuit above corresponds to $X \otimes X$ followed by CNOT, and the matrix representation of the circuit is just the product of the matrices representing each gate. Using the matrix representation of $X \otimes X$ from part (a), we find

$$\begin{pmatrix} 0 & X \\ X & 0 \end{pmatrix}\begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} 0 & X^2 \\ X & 0 \end{pmatrix} = \begin{pmatrix} 0 & I \\ X & 0 \end{pmatrix}$$

In the same manner as the $X_2X_1$ gate, we can write the matrix representation of $X_2I_1$ as

$$X \otimes I = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}$$

The matrix representing the above circuit is thus

$$\begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}\begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} = \begin{pmatrix} 0 & I \\ X & 0 \end{pmatrix}$$

We see that the two matrices agree, verifying that the circuits are equal. In the language of linear algebra, $(X \otimes X)\text{CNOT} = \text{CNOT}(X \otimes I)$.

## Problem 3: Controlled logic

One of the easiest ways to create correlations between bits is through conditional gates. Let $M$ be a gate which acts on a single bit. We denote by C$M$ the gate acting on two bits that applies $M$ to bit 0 if bit 1 is 1 and does nothing otherwise. Write down the matrix for C$M$. When is this matrix invertible?

### Solution

Without knowing anything about the operator $M$, we know that $CM : (\vec{e}_0, \vec{v}) \to (\vec{e}_0, \vec{v})$ and $CM : (\vec{e}_1, \vec{v}) \to (\vec{e}_1, M\vec{v})$. Looking at the matrix

$$\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}$$

We can verify that it performs the correct operation by exploiting the Kronecker product. If $b_1$ is in the zero state and $b_2$ is in the arbitrary state $\vec{v}$, then the resultant joint state is specified using the Kronecker product:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \vec{v} = \begin{pmatrix} \vec{v} \\ 0 \end{pmatrix}$$

Similarly, if $b_1$ is in the state one, then the joint state is described by

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \vec{v} = \begin{pmatrix} 0 \\ \vec{v} \end{pmatrix}$$

Then, we can check that the matrix $M$ performs the desired operation:

$$\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}\begin{pmatrix} \vec{v} \\ 0 \end{pmatrix} = \begin{pmatrix} \vec{v} \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \vec{v}$$

$$\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}\begin{pmatrix} 0 \\ \vec{v} \end{pmatrix} = \begin{pmatrix} 0 \\ M\vec{v} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes M\vec{v}$$

In the first case, $b_0$ is left in the state $\vec{v}$, and in the second case it is left in the state $M\vec{v}$.

# Problem 4: Random box

Consider a black box with a single input bit that it flips with probability $p$ and leaves the bit unchanged with probability $1-p$.

(a) Write down the matrix for this operation. Is it invertible? Is the inverse guaranteed to produce a valid output distribution?

(c) Is it possible to construct a valid gate that maps $\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix}$?

This exercise will show that if $p = \frac{1}{2}$, then there is no gate that "unscrambles" the completely random state. For any $p > 0$, the inverse is not a valid gate, so there is no possible gate that returns the distribution back to the original. Remarkably, such an operator does exist in quantum computing!

## Solution

Let $\vec{v}$ be an arbitrary single-bit state. We want our operation $M$ to have the property that

$$M\vec{v} = (1-p)\vec{v} + pX\vec{v}$$

It is sufficient to ensure this property for any basis, so we choose $\vec{e}_0, \vec{e}_1$. For these vectors, $\vec{e}_0 \mapsto (1-p)\vec{e}_0 + p\vec{e}_1$ and $\vec{e}_1 \mapsto (1-p)\vec{e}_1 + p\vec{e}_0$. The matrix that performs this operation is

$$\begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} = (1-p)I + pX$$

This gives a probabilistic interpretation of the gates themselves–a linear combination of two gates corresponds to applying one gate with some probability and another gate with some probability. The gate set also forms a vector space!

Writing down the inverse of this matrix, we have

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

When $p < 1/2$, the prefactor is positive, and $-p \leq 0$. When $p > 1/2$, the prefactor is negative, and $-(1-p) \leq 0$. This shows that the inverse is not a valid gate unless $p = 0$, because we do not consider gates that can produce negative probabilities. Furthermore, when $p = \frac{1}{2}$, the matrix is singular. This is reflective of the fact that a maximally random distribution is equally likely to correspond to any initial state.

This is the intuition underlying the question about constructing an operation that maps $\frac{1}{2}(\vec{e}_0 + \vec{e}_1) \mapsto \vec{e}_0$. If $M$ represents a valid gate, then

$$M = \begin{pmatrix} a & b \\ 1-a & 1-b \end{pmatrix}$$

We have to have $a, b \leq 1$ for the gate to be valid and $\frac{1}{2}(a+b) = 1$ for it to perform the correct operation. This gives

$$M = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

This gate is incredibly uninteresting, because it maps any input state to 0. It is certainly not invertible.

One aspect separating quantum probability from classical probability is that there are many invertible operations that produce completely random states but are invertible. This is an affect of quantum superpositions!

# Problem 5: Limitations of classical probability

Remember that valid gates must map positive, normalized distributions to positive, normalized distributions.

(a) What is the most general single-bit gate you can write down?

(b) What subset of these operations are invertible?

This exercise shows that we are basically limited to unsigned permutation matrices with reversible probabilistic computation. As we discussed, all classical operations can be made reversible, so this completely characterizes the available gates. This is eminently reasonable, because the unsigned permutation matrices characterize all binary maps! As we will see later, quantum computing allows for much more freedom in constructing gates.

### Solution

The most general single-bit gate $M$ is one that maps valid probability distributions to other valid distributions. By considering the basis vectors $\vec{e}_0, \vec{e}_1$, we can see that it is a necessary condition for the columns of this matrix to be normalized, positive distributions over the two states:

$$M = \begin{pmatrix} a & b \\ 1-a & 1-b \end{pmatrix}$$

Furthermore, if we have an arbitrary state $\vec{v}$, then we can write $\vec{v} = c\vec{e}_0 + d\vec{e}_1$ where $c + d = 1$. Then

$$M\vec{v} = (ca + db)\vec{e}_0 + (c(1-a) + d(1-b))\vec{e}_1$$

and $ca + db + c(1-a) + d(1-b) = c + d = 1$. This shows that the condition is also sufficient. Inverting $M$ gives

$$M^{-1} = \frac{1}{\det M}\begin{pmatrix} 1-b & -b \\ a-1 & a \end{pmatrix}$$

Since $a \leq 1$, $a - 1 < 1$. If $\det M > 0$, then the upper right-hand element is negative if $b \neq 0$. and the lower left-hand element is negative if $a \neq 1$, so it must be the case that $b = 0$ and $a = 1$ for this to be a valid operation. Similarly, if $\det M < 0$ then $a = 0$ and $b = 1$. This shows that $M = X$ is the only valid invertible probabilistic single-bit gate.

*Note:* The set of problems that are solvable in polynomial time by a probabilistic turing machine is called BPP (bounded-error probabilistic polynomial-time). BPP is a subset of BQP (bounded-error quantum polynomial-time), but it is currently an open question whether it is a proper subset or they are equal. Nevertheless, this example demonstrates something exciting about quantum computing. Classical operations can only make states more random. As we will see, quantum operations leverage superposition and interference to extend invertible operations to a wider class of distributions.

## Problem 6 (Addtl): Information Entropy

Consider a single bit $\vec{b} = \big(p, 1-p\big)$, and define the Shannon entropy to be $H(\vec{b}) = -p\log_2(p) - (1-p)\log_2(1-p)$. The Shannon entropy quantifies the randomness in a distribution, with a larger entropy indicating a more random distribution. Let $M$ be a valid single-bit gate. Show that $H(M\vec{b}) \geq H(\vec{b})$.