Transferring Entanglement Between Different Dimensions

Timothy Forrer
Level 4 Project, MSci Natural Sciences
Supervisor: Dr V. Kendon
Department of Physics, Durham University

Submitted: April 2, 2021

Abstract

Quantum random walks have been an area of significant interest over the past two decades within quantum computation and information. They exhibit many desirable qualities that aid quantum computations and are extremely versatile for use in the design of quantum algorithms. In this report I analyse a recent publication that uses quantum walk dynamics in order to generate higher dimensional entangled states.

Contents

1 Introduction

Quantum computing is an area of intense active research, with its sensational results often making it into mainstream media. Loosely speaking, it is a field that seeks to examine how quantum phenomena can be exploited to allow for greater and more powerful computations than currently possible using classical computers. Many schemes for quantum computation utilise qubits, the quantum analogue of the classical bit which is the unit of information in classical computing. The main difference between the bit and the qubit, is that the bit can only exist in one of two states at any one time, whereas the qubit, due to it's quantum nature, can exist in superposition of both the possible states. Other models utilising qudits, which exist in superposition of d states rather than just two, have been proposed and they can unlock certain advantages at the cost of being more complex to implement.

In addition to superposition, another well known quantum phenomena that is often taken advantage of in quantum computing is *entanglement*, correlations present in quantum systems that are far stronger than possible to find in classical systems. There are a variety of protocols that require the presence of entangled qubits in order to achieve results not possible with classical computers, for example superdense coding [1], quantum key distribution [2] and quantum teleportation [3]. Higher dimensional entanglement, that is entanglement between qudits, is able to unlock even further benefits in quantum algorithms and as such its generation is extremely important but comes with its own challenges.

Quantum walks are powerful tools in the landscape of quantum computing. Much like their classical analogues, they exhibit many properties that are desirable for computations and are an extremely useful building block for many algorithms designed for quantum computers [4]. Specific research interest into the quantum variant stems from their very significant divergences from the classical, including different spreading speeds and quantum correlations, known as entanglement, that have no classical comparison. Their power is such that quantum walks can simulate any quantum computation and therefore are a model for universal quantum computing [5]. There is also evidence of robust performance even when the quantum computer is not perfectly isolated from its environment, and in certain situations it has been shown that decoherences due to interactions with the environment is beneficial for a given computation [6]. Quantum walks are divided into two categories, *discrete* and *continuous* time, these labels describing the nature of the evolution of the walker as the quantum walk progresses. Continuous time quantum walks have been shown to solve a wide range of problems in a number of different settings, in some cases exponentially faster than a classical computer is able to [7]. However, the focus of this report will be on entangled state generation and transfer which requires more than one subspace in our system, a setting which lends itself much more readily to discrete time quantum walks.

A potential solution to the demanding task of generating higher dimensional entanglement has been proposed [8] which uses the dynamics of quantum walks to transfer lower dimensional entanglement between qubits, which is far simpler to generate, into the high dimensional qudits. Whilst this scheme can be used to some moderate degree of success, I will present an alternative devised to operate in a similar setting but utilising ancilla-based quantum computing to instead transfer entanglement optimally.

In this report, a primer on entanglement, and two models of quantum computing, quantum walks and ancilla-based quantum computing, is given in §2 Following this, §3 will focus on the protocol that uses quantum walk dynamics to facilitate the transfer of entanglement, in particular analysing it's efficiency in achieving the aim of entanglement transfer. The ancilla-based quantum computing scheme for entanglement transfer is presented in §4, and comparison is also given highlighting the advantages this scheme has over the quantum walk based protocol. Further uses of the AQC scheme are also presented in this section.

Finally, a short summary is presented in §5.

MOVE TO WHEN NEEDED As mentioned above, discrete time quantum walks are much more suited for the purposes of this report, therefore future references to quantum walks will be assumed to be the discrete variant unless stated otherwise. We will also use QW to denote (discrete) quantum walk. The mathematical notation used in this report follows standard conventions, in particular it should be made clear the equivalency between $|u_1, u_2\rangle \equiv |u_1\rangle |u_2\rangle \equiv |u_1\rangle \otimes |u_2\rangle$, where all forms will be used interchangeably. MOVE TO WHEN NEEDED

2 Background

2.1 Quantum Computation

2.1.1 Mathematical Framework

When describing a qubit, we say that it lives in a Hilbert space $\mathcal{H}$ spanned by two states, labelled $|0\rangle$ and $|1\rangle$ in the *computational basis*. Naturally we can also use other bases for describing our qubit state. A common alternative is the basis $\{|+\rangle, |-\rangle\}$, where

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

$$|-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \tag{2}$$

In order to describe a collection of n qubits together, we can take the tensor product of each of the Hilbert spaces that each qubit resides in to form a 2^n dimensional Hilbert space

$$\mathcal{H}_{2^n} = \mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \cdots \otimes \in_2 \tag{3}$$

$$= \bigotimes_{i=0}^{2^n-1} \mathcal{H}_2 \tag{4}$$

2.1.2 Gates and Circuits

The circuit notation used in describing quantum circuit schematics is derived from circuits used in classical computation. Straight lines in our circuit represent qubits (rather than bits

Operator	Gate
(Pauli) X	$-\boxed{X}-$
(Pauli) Z	$-\boxed{Z}-$
Hadamard	$-\boxed{H}-$
QFT	$-\boxed{F}-$

Table 1: Table of operators and their gate representations in quantum circuits.

as in classical circuits). We represent operations on these qudits via gates placed on the straight lines corresponding to the qudits we wish to act on. A table of gates relevant for this report is given in Table 1.

When reading a circuit, time flows from left to right and a gate must finish its operation on a qudit before the next gate is operated. However, in the case of two seperate qudits the ordering of gates that do not interact between the two of them is irrelevant since they commute, as discussed in subsection 2.1.1.

2.2 Entanglement

2.2.1 Measuring entanglement

There are many different methods of quantifying entanglement, and there is no one standard method for doing so as different measures are preferable in different scenarios. Here we will introduce and define two common measures of entanglement.

We first define *von Neumann entropy*, S, used for pure states.

$$S(\rho) = -\mathrm{tr}(\rho \ln \rho), \tag{5}$$

where ρ is the density matrix representing the quantum system and ln is the natural matrix logarithm. It is often easier to compute this quantity using the spectral decomposition of ρ,

$$\rho = \sum_a \lambda_a |a\rangle \langle a| \,. \tag{6}$$

Using this form we can compute the entropy by

$$S(\rho) = -\sum_a \lambda_a \ln \lambda_a. \tag{7}$$

We define the *ebit* as is the unit of bipartite entanglement. It is the entanglement contained in a Bell State [9]. Maximally entangled states in $d \times d$ dimensions have $log_2(d)$ ebits.

Whilst QW dynamics are unitary, which ensures that pure states retain their purity as the walk progresses, we will see that this measure of entanglement is not appropriate for measuring how well our entanglement has been transferred as our protocol will also involve projections, which are not unitary. To this end, we introduce an alternative measure of entanglement called *negativity,* $\mathcal{N}$ [10].

$$\mathcal{N}(\rho) = \frac{||\rho^{\Gamma_A}||_1 - 1}{2}, \tag{8}$$

where ρ^{Γ_A} denotes the *partial transpose* of ρ with respect to the subsystem $\mathcal{H}_A$ and $||X||_1 = Tr(\sqrt{X^\dagger X})$ denotes the *trace norm* of X.

Similarly to von Neuman entropy, it is possible to rewrite this definition in terms of eigenvalues, this time eigenvalues of ρ^{Γ_A},

$$\mathcal{N}(\rho) = \sum_{\lambda_a < 0} |\lambda_a| = \sum_{\lambda_a} \frac{|\lambda_a| - \lambda_a}{2}. \tag{9}$$

2.3 Random Walks

2.3.1 Classical Random Walk

Before discussing quantum random walks, I will first motivate their design using the example of a classical random walk on a discrete number line. In the classical random walk, a walker (often described as being somewhat inebriated) is constrained to moving up and down a discrete number line, starting their walk at the origin. To determine whether to take a step to the left (-1) or the right (+1), the walkers flips an unbiased coin, moving to the right if the coin lands on heads and to the left if the coin lands on tails. The process is can be repeated over and over until a desired stopping point is reached, after a given number of coin flips, or until the walker has reached a specific destination, such as their house in the case of the inebriated walker. The walk can also continue on forever and in this limit, the walker will reach every point on the number line. The probability distribution describing the probability of the walker being in any given position away from the origin is given by a binomial distribution, shown in orange in Figure 1 for 100 coin flips.

2.3.2 Quantum Random Walk

With the classical random walk model in mind, we are now in a position to 'quantise' it into the *quantum random walk*. In our walker system, we can divide the overall Hilbert space of the QW, $\mathcal{H}$, into two subspaces, the coin subspace $\mathcal{H}_C$ and the position subspace of the walker $\mathcal{H}_W$.

$$\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_W. \tag{10}$$

We note that whilst we do not place any constraints on the size of $\mathcal{H}_W$, we choose $dim(\mathcal{H}_C) = 2$, which is the natural thing to do when quantising our classical walk since the classical coin has two possible states. To aid distinguishability between coin states and position states, we write that

$$\langle \mathcal{H}_C \rangle = \{ |\uparrow\rangle , |\downarrow\rangle \} \tag{11}$$
$$\langle \mathcal{H}_W \rangle = \{ |k\rangle \, | k \in \mathbb{Z} \}, \tag{12}$$

where $\langle U \rangle$ denotes a set of vectors which span U. Therefore, the states $|\uparrow\rangle , |\downarrow\rangle$ take the place of heads and tails on our quantum 'coin'. Having defined the Hilbert space within which the walk will be conducted, we can now define operators within our space that will dictate how the QW will proceed. We first define the 'coin flip' operator $C \in \mathcal{H}_C$. There are several choices for C, details of which can be found here [11]. As detailed in [11], for walks on a line, if we restrict ourselves to choosing an unbiased coin with real coefficients the Hadamard coin

$$C = \frac{1}{\sqrt{2}} \Big[\, |\uparrow\rangle \langle\uparrow| + |\uparrow\rangle \langle\downarrow| + |\downarrow\rangle \langle\uparrow| - |\downarrow\rangle \langle\downarrow| \, \Big] \tag{13}$$
$$= \frac{1}{\sqrt{2}} \Big[(|\uparrow\rangle + |\downarrow\rangle) \langle\uparrow| + (|\uparrow\rangle - |\downarrow\rangle) \langle\downarrow| \, \Big] \tag{14}$$

is the only choice of coin available. Equation (5) makes obvious the action of C; if the coin state is $|\uparrow\rangle$ then it becomes an equal superposition of $|\uparrow\rangle + |\downarrow\rangle$, if the coin state is in $|\downarrow\rangle$ then we get an equal superposition of $|\uparrow\rangle - |\downarrow\rangle$. These two equal superpositions are often denoted as $|+\rangle$ and $|-\rangle$ respectively.
We then define our shift operator $S \in \mathcal{H}$ which allows the position of our walker to change, dependent on the state of the coin.

$$S = \sum_k |\uparrow\rangle \langle\uparrow| \otimes |k+1\rangle \langle k| + |\downarrow\rangle \langle\downarrow| \otimes |k-1\rangle \langle k| \, . \tag{15}$$

Again, this representation of S makes manifest its effect on our walker. If the coin is in the $|\uparrow\rangle$, then we take a step in the +1 direction, if in the $|\downarrow\rangle$ then we take a step in the -1 direction. The probability distribution of such a walk is plotted in Fig 1, where the initial coin state is $|\downarrow\rangle$, and is compared to a classical random walk.

Whilst the above choice of S is the most common on the number line, it is also possible to define an alternative choice of shift operator,

$$\tilde{S} = \sum_k |\uparrow\rangle \langle\uparrow| \otimes |k\rangle \langle k| + |\downarrow\rangle \langle\downarrow| \otimes |k+1\rangle \langle k| \, . \tag{16}$$

Th subtle difference between $\tilde{S}$ and S is that $\tilde{S}$ can only move in the +1 direction of the number line and has no 'left moving' part, so to speak. This means that $\tilde{S}$ is restricted to the non-negative integers and unlike S, can occupy all $|x\rangle$ for $0 \leq x \leq T$, where T is the number of time steps in our QW.

2.4 Ancilla-Based Quantum Computing

Quantum walks is but one of many universal quantum computing models. Another model is known as *ancilla-based quantum computing*, which, in particular, aims to resolve two conflicting demands when building quantum computers: We wish for our qubits to be well

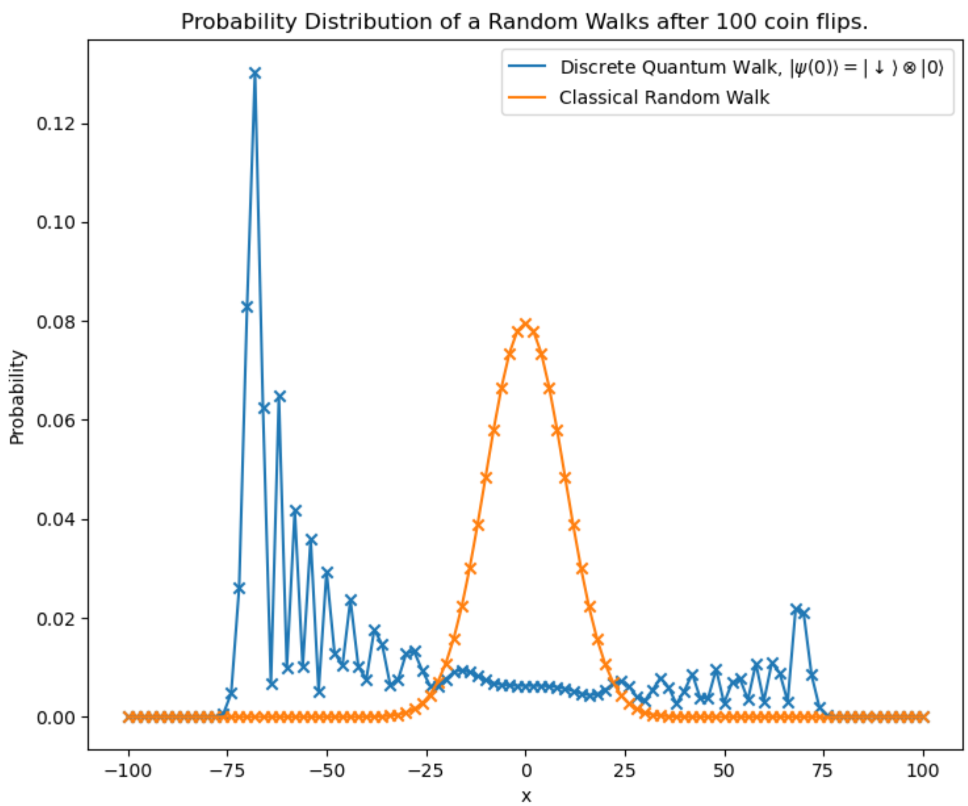

Figure 1: A comparison between the classical and quantum walk on a line. The quantum walk is initially in the $|{\downarrow}\rangle$ state in the coin subspace of the walk.

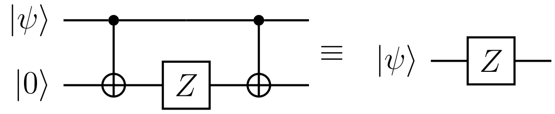

Figure 2: Two circuits that implement a Z gate acting on an arbitrary qubit $|\psi\rangle$.

isolated to prevent decoherence, yet also want them to interact with each other to perform our quantum computations. However, a qubit cannot distinguish between unwanted and wanted interactions, resulting in the need for a balancing act that adequetely addresses these two issues.

Ancilla-based quantum computing aims to resolve this conflict by using additional qubits, known as ancilla qubits, which mediate interactions between the main register qubits. Using this model, it is possible to implement two complementary registers of qubits. The main register can be designed to be strongly isolated from interactions, bar a select few natural interactions with the ancilla registry, whose qubits can be easily manipulated but decohere much faster. Quantum computations can be performed by *delocalising* quantum information from the main register across both the register and the ancilla. After performing a computation on the ancilla qubit, the quantum information is then *relocalised* back into the main register. The ancilla can be reset to the initial state and used again for other computations. A simple example of a circuit designed for ancilla-based quantum computing is shown in Figure 2.

The circuit on the left hand side of Figure 2 does not directly implement any unitary transformations on an arbitrary qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, but instead indirectly acts on $|\psi\rangle$ via the ancilla qubit, to which the quantum information of $|\psi\rangle$ is shared to via a CNOT gate. Acting a Z gate on the ancilla and relocalising the quantum information with another CNOT gate 'passes on' the effects of the Z gate.

Later in this report I will highlight how this particular model of quantum computing is well suited for the protocol outlined in [8], which will be discussed in detail in the following section.

3 Entanglement Transfer using Quantum Walks

We now review the protocol presented in [8] which utilises quantum walk dynamics in order to generate higher dimensional entanglement.

In this section we use following notation:

- $|u\rangle^{(i)}$ is a state belonging to the subspace $\mathcal{H}^{(i)} = \mathcal{H}_C^{(i)} \otimes \mathcal{H}_W^{(i)}, i \in \{1, 2\}$.
- $|u\rangle_J$ is a state belonging to the subspace $\mathcal{H}_J = \mathcal{H}_J^{(1)} \otimes \mathcal{H}_J^{(2)}, J \in \{C, W\}$.
- $|u\rangle_J^{(i)}$ is a state belonging to the subspace $\mathcal{H}_J^{(i)}$.

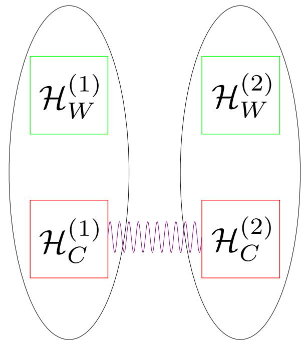

Figure 3: The initial prepared state has entanglement solely between the two coin subspaces. Figure is an edited version from FIG 3 from [8].

In this mathematical framework the overall Hilbert space of the quantum system is comprised of two quantum walk subspaces,

$$\mathcal{H} = \mathcal{H}^{(1)} \otimes \mathcal{H}^{(2)} \tag{17}$$

$$= \mathcal{H}_C^{(1)} \otimes \mathcal{H}_W^{(1)} \otimes \mathcal{H}_C^{(2)} \otimes \mathcal{H}_W^{(2)}. \tag{18}$$

The basic premise of this protocol is this:

1. Entangle the two coin spaces of the walkers $\mathcal{H}_C^{(i)}$. (Fig 3.)
2. Proceed with the random walk.
3. Use a projection $\mathcal{P}_\gamma = |\gamma\rangle\langle\gamma|, |\gamma\rangle \in \mathcal{H}_C^{(1)}$ to then transfer the entanglement so that it solely exists in the subspace $\mathcal{H}_W^{(1)} \otimes \mathcal{H}_C^{(2)} \otimes \mathcal{H}_W^{(2)}$.
4. In similar fashion, find a projection $\mathcal{P}_\delta = |\delta\rangle\langle\delta|, |\delta\rangle \in \mathcal{H}_C^{(2)}$ to transfer the entanglement to exist between the two walker subspaces, $\mathcal{H}_W^{(i)}$, only.
5. Accumulate entanglement in the walker subspaces by once more entangling the two coin spaces and repeating the protocol.

In this way, we are able to generate arbitrary amounts of high dimensional entanglement. As is the case with many quantum walk based protocols, particular attention must be paid to the choice of coin used for the quantum walk, as it will have a large impact on the success of the protocol. In the presentation of the protocol in [8], the shift operator $\tilde{S}$ (the operator with no left moving part) is used, and as such we will too present this overview using the same shift operator. It should be noted however, that the choice of shift operator, S or $\tilde{S}$, has no real impact upon the workings of this protocol.

3.1 Transfer using identity coin operator

We first use the example of a quantum walk with coin I, the identity. We prepare a state $|\psi(0)\rangle$ with the coin states entangled and walkers at the origin

$$|\psi(0)\rangle = \underbrace{\frac{1}{\sqrt{2}}\Big[\,|\uparrow\rangle_C^{(1)}\,|\uparrow\rangle_C^{(2)} + |\downarrow\rangle_C^{(1)}\,|\downarrow\rangle_C^{(2)}\,\Big]}_{\text{Bell State}} \otimes |0\rangle_W^{(1)}\,|0\rangle_W^{(2)}\,. \tag{19}$$

Following this we apply our coin, I, and then use our shift operator S to advance the quantum walk. Explicitly (dropping the indices and combining some of our kets together) we obtain

$$|\psi(1)\rangle = \tilde{S}I\,|\psi\rangle = \frac{1}{\sqrt{2}}\Big[\,|\uparrow,\uparrow\rangle\,|0,0\rangle + |\downarrow,\downarrow\rangle\,|1,1\rangle\,\Big]. \tag{20}$$

We then project the part of $|\psi(1)\rangle$ residing in the $\mathcal{H}_C^{(1)}$ subspace onto the vector $|\gamma\rangle$, using the projective operator $\mathcal{P}_\gamma = |\gamma\rangle\langle\gamma| \in \mathcal{H}_C^{(1)}$. Choose $|\gamma\rangle = \frac{1}{\sqrt{2}}\big[\,|\uparrow\rangle + |\downarrow\rangle\,\big]$ which then gives us

$$\mathcal{P}_\gamma\,|\psi(1)\rangle = \frac{1}{2}\Big[\,|\gamma\rangle \otimes \big(\,|\uparrow,0,0\rangle + |\downarrow,1,1\rangle\,\big)\Big]. \tag{21}$$

Similarly, we project the other walker subspace to $|\delta\rangle$ which we can in this instance take to be the same state as $|\gamma\rangle = \frac{1}{\sqrt{2}}\big[\,|\uparrow\rangle + |\downarrow\rangle\,\big]$,

$$\mathcal{P}_\delta\mathcal{P}_\gamma\,|\psi(1)\rangle = \frac{1}{2\sqrt{2}}\Big[\,|\gamma\rangle \otimes |\delta\rangle \otimes \frac{1}{\sqrt{2}}\Big(\,|0,0\rangle + |1,1\rangle\,\Big)\Big]. \tag{22}$$

Renormalising we see that we have the state

$$|\gamma\rangle_C^{(1)} \otimes |\delta\rangle_C^{(2)} \otimes \underbrace{\frac{1}{\sqrt{2}}\Big[\,|0,0\rangle + |1,1\rangle\,\Big]_W}_{\text{Bell State}}, \tag{23}$$

which has a Bell State in the $\mathcal{H}_W$ subspace, and the states in $\mathcal{H}_C$ are separable. Therefore we have transferred the entanglement that originally resided in the coin subspace to the walker one.

3.2 Accumulation

Of course the real interest in this protocol, as previously stated, is the ability to accumulate the entanglement transferred from the lower dimensional coin subspace to the higher dimensional walker one. We now demonstrate how we can do this, again using I as our coin. We start with the final state obtained in the previous section (equation (22)) and re-entangle the coin subspaces, redefining the resulting state as our $|\psi(0)\rangle$,

$$|\gamma\rangle_C^{(1)} \otimes |\delta\rangle_C^{(2)} \otimes \frac{1}{\sqrt{2}}\Big[\,|0,0\rangle + |1,1\rangle\,\Big]_W \tag{24}$$

$$\xrightarrow{\text{Entangle } \mathcal{H}_C} \frac{1}{\sqrt{2}}\Big[\,|\uparrow,\uparrow\rangle + |\downarrow,\downarrow\rangle\,\Big]_C \otimes \frac{1}{\sqrt{2}}\Big[\,|0,0\rangle + |1,1\rangle\,\Big]_W \tag{25}$$

$$= |\psi(0)\rangle\,. \tag{26}$$

We then proceed with the walk, however taking two steps instead of one this time,

$$|\psi(2)\rangle = (\tilde{S}I)^2 \, |\psi(0)\rangle \tag{27}$$

$$= \frac{1}{2}\Big[\, |\uparrow,\uparrow\rangle \, \big(\, |0,0\rangle + |1,1\rangle \, \big) + |\downarrow,\downarrow\rangle \, \big(\, |2,2\rangle + |3,3\rangle \, \big) \Big]. \tag{28}$$

We use the same projectors in the two coin subspaces, $\mathcal{P}_\gamma \in \mathcal{H}_C^{(1)}, \mathcal{P}_\delta \in \mathcal{H}_C^{(2)}$, and after renormalisation have the final state

$$|\gamma\rangle \otimes |\delta\rangle \otimes \frac{1}{2}\Big[\, |0,0\rangle + |1,1\rangle + |2,2\rangle + |3,3\rangle \, \Big]. \tag{29}$$

In this way we have now accumulated two ebits in the walker subspace, and can continue to repeat this process to accumulate arbitrarily large amounts of entanglement into our walker subspace, noting that each n^{th} iteration requires an additional 2^{n-1} steps in our quantum walk.

3.3 Drawbacks

In the example given above, which is the example given in [8], we have demonstrated that this protocol can transfer all of the entanglement to our qudits. However we are left with a crucial problem in that the example does not in fact use quantum walk dynamics to transfer the entanglement, seeing as there effectively is no coin driving the walk. In analysing the protocol with the Hadamard coin, I found that only one ebit of entanglement could be transferred efficiently and subsequent repetitions of the protocol to store further amounts of entanglement could not be done optimally. Simulations of the protocol found that at most around 1.6 ebits of entanglement can be stored with a Hadamard QW, as shown in Figure X. Furthermore, the projective measurements employed as part of the protocol mean that it is not unitary and is non trivial to reverse in order to retrieve the entanglement out from the entangled qudits. The experimental implementation suggested in the paper also required the use of post selection, where undesirable states were discarded and the protocol run again. All this in combination results in a protocol which is rather inefficient in achieving its aims. However, the example clearly shows that it is possible to efficiently transfer entanglement to qudits, but highlights that a different quantum computational model should perhaps be employed. In the §4, an alternative protocol in the ancilla-based quantum computing model is proposed which solves most of the inefficiencies present in this QW based proposal.

4 Entanglement Transfer using Ancilla-Based Quantum Computing

I will now outline an alternative entanglement transfer scheme designed to operate in the same setting at the QW based protocol, but instead is able to transfer entanglement deterministically and also retrieve it perfectly as well.

A circuit schematic is given in Figure 4, and essentially is two copies of the same circuit.

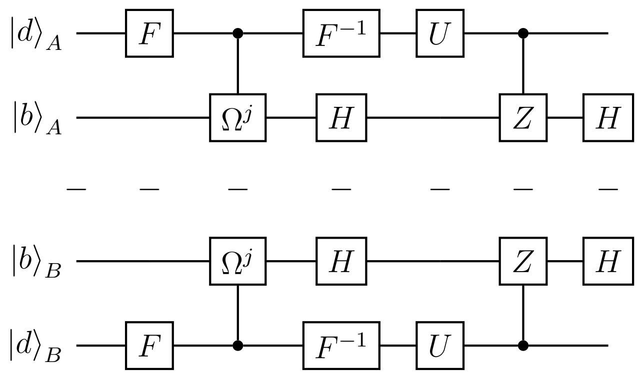

Figure 4: The AQC circuit for entanglement transfer.

Since the two halves of the circuit are completely seperate from each other, this circuit is suitable for the same experimental setup described for the QW protocol, where it is imagined that there are two labs which are spatially seperated, and have a shared source of Bell state entangled qubits which are sent individually to each lab. The F gate is one that acts the Quantum Fourier Transform on the qudit basis states,

$$F|x\rangle = \frac{1}{2^{\frac{d}{2}}}\sum_{y=0}^{d-1}\omega^{xy}|y\rangle, \tag{30}$$

where $\omega = e^{i\frac{2\pi}{d}}$. The matrix representation of F in the computational basis is given by by the $d \times d$ matrix

$$F = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{d-2} & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(d-2)} & \omega^{2(d-1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \dots & \omega^{(d-2)(d-1)} & \omega^{(d-1)^2} \end{pmatrix} \tag{31}$$

The gate Ω is given by

$$\Omega = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}, \tag{32}$$

therefore the controlled-Ω gates are ones that act as

$$C\text{-}\Omega^j(|x\rangle_d|y\rangle_2) = |x\rangle_d \otimes \Omega^{xj}|y\rangle_2 \tag{33}$$
$$= |x\rangle_d \otimes \omega^{xyj}|y\rangle_2 \tag{34}$$

where $|x\rangle_d$ is the control qudit and $|y\rangle_2$ the target qubit.

Proposition 1. *In the Fourier basis, the C-Ω^j operator acts on the product state $|+_k\rangle_d \otimes |1\rangle_2$ to give $|+_{k+j}\rangle_d \otimes |1\rangle_2$*

Proof. The Fourier basis state $|+_k\rangle_d$ is given by

$$|+_k\rangle_d = \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{km} |m\rangle . \tag{35}$$

Therefore

$$\begin{aligned} C\text{-}\Omega^j \left(|+_k\rangle_d |1\rangle_2\right) &= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} C\text{-}\Omega^j \left(\omega^{km} |m\rangle_d \otimes |1\rangle_2\right) && (36) \\ &= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{mj} \left(\omega^{km} |m\rangle_d \otimes |1\rangle_2\right) && (37) \\ &= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{(k+j)m} |m\rangle_d \otimes |1\rangle_2 && (38) \\ &= \left(\frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{(k+j)m} |m\rangle_d\right) \otimes |1\rangle_2 && (39) \\ &= |+_{k+j}\rangle_d \otimes |1\rangle_2 . && (40) \end{aligned}$$

□

Proposition 1 highlights the somewhat counterintuitive nature of the circuit, and indeed the counterintuitive way ancilla-based quantum computing can exploit quantum phenomena. We act with a controlled operation, where the qudit is the control and the qubit is the target, yet it is the qudit state that is changed and the qubit is left ‘untouched’. This idea is crucial for this proposal to achieve the desired outcomes, as we are required to shift the qudit states, but are unable to do so without being in the Fourier basis, where delocalised phase changes shift the Fourier basis states. Analogously to the QW protocol, the qudit can be thought of as a walker driven by the ancilla qubit ‘coin’.

4.1 Examples

To explicitly demonstrate how the circuit operates, an example of storing two Bell states and retrieving them is given in the following three subsections.

4.1.1 Storage

4.1.2 Accumulation

4.1.3 Retrieval

Given that this is a circuit that solely utilises unitary transformations, retrieval of entangled Bell pairs is trivially done by running the circuit backwards. Furthermore, there is no requirement to use the same ancilla qubits to retrieve the entanglement. By the end of the circuit the ancilla qubits are in the $|0\rangle$ state, therefore any pair of ancilla qubits in the $|0\rangle$ state may be used to retrieve the entanglement.

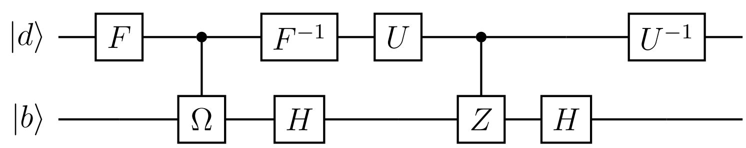

Figure 5: The AQC circuit with one qudit and qubit which can store arbitrary qubit states in the qudit.

4.2 Further uses of the circuit

Although the goal of this research was to design a protocol in a similar setting to the QW based protocol that outperformed the QW protocol, further testing of the proposed circuit showed that it had further uses beyond storage of entanglement. In this section, we consider the setting where we are in one lab and have one qubit and one qudit, and therfore only have one half of the total circuit. By adding one additional gate which uncomputes U, as shown in Figure 5, we can in fact store arbitrary qubit states in our qudits. Furthermore, if more than one qubit is stored in the circuit then it is in fact possible to retrieve the qubit states in a different order to that which they were stored, as long as the order in which they were stored is known. Therefore, this circuit can be utilised in turning qudits into quantum random access memory.

4.2.1 Quantum Random Access Memory

The procedure for utilising the circuit as a quantum random access memory is as follows. Assume that there is a qudit in state $|0\rangle_d$ and two qubits, qubit 1 and qubit 2, in states $a|0\rangle_2 + b|1\rangle_2$ and $c|0\rangle_2 + d|1\rangle_2$ respectively. We first run the circuit with the qudit and and qubit 1,

$$|0\rangle_d \otimes (a|0\rangle + b|1\rangle) \longrightarrow (a|0\rangle_d + b|1\rangle_d) \otimes |0\rangle_2 . \tag{41}$$

We then replace qubit 1 with qubit 2 and run the circuit again,

$$(a|0\rangle_d + b|1\rangle_d) \otimes (c|0\rangle_2 + d|1\rangle_2) \longrightarrow (ac|0\rangle_d + bc|1\rangle_d + ad|2\rangle_d + bd|3\rangle_d) \otimes |0\rangle_2 . \tag{42}$$

In order to retrieve qubit 2 back, this can be done simply by running the circuit backwards. However, if we wish to retrieve the qubit 1 state then we first require a specific unitary operator. The form of the unitary transform can be found as follows. Rewrite each of the qudit basis state numbers in binary, i.e.

$$|0\rangle = |00\rangle \tag{43}$$
$$|1\rangle = |01\rangle \tag{44}$$
$$|2\rangle = |10\rangle \tag{45}$$
$$|3\rangle = |11\rangle . \tag{46}$$

Rewriting the final state of (42) in this way we obtain the following expression

$$ac|0\rangle + bc|1\rangle + ad|2\rangle + bd|3\rangle = ac|00\rangle + bc|01\rangle + ad|10\rangle + bd|11\rangle , \tag{47}$$

which can be rewritten as the product state

$$\underbrace{(c|0\rangle + d|1\rangle)}_{\text{Qubit 2}} \otimes \underbrace{(a|0\rangle + b|1\rangle)}_{\text{Qubit 1}}. \tag{48}$$

Quite remarkably the qudit has an intuitive alternative expression as the product state of qubit 1 and qubit 2. In order to retrieve qubit 1 by running the circuit backwards, we need to instead be able to express the qudit state as

$$\underbrace{(a|0\rangle + b|1\rangle)}_{\text{Qubit 1}} \otimes \underbrace{(c|0\rangle + d|1\rangle)}_{\text{Qubit 2}}. \tag{49}$$

This can be thought of as switching the positions of our two qubits, which leads us to the unitary transformation we need. We need a map, M, which will flip the positions of our two qubits. M is found by mapping each of the binary qudit state representations $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ to the basis state with the binary digits switched.

$$|00\rangle \mapsto |00\rangle \tag{50}$$
$$|01\rangle \mapsto |10\rangle \tag{51}$$
$$|10\rangle \mapsto |01\rangle \tag{52}$$
$$|11\rangle \mapsto |00\rangle. \tag{53}$$

We can check that this gives us the form that we want taking the RHS of (47) and acting M on it to obtain

$$M(ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle) = ac|00\rangle + ad|10\rangle + bc|01\rangle + bd|11\rangle \tag{54}$$
$$= \underbrace{(a|0\rangle + b|1\rangle)}_{\text{Qubit 1}} \otimes \underbrace{(c|0\rangle + d|1\rangle)}_{\text{Qubit 2}}, \tag{55}$$

as required. M can be expressed in terms of the original qudit state labelling by converting the binary back

$$|0\rangle \mapsto |0\rangle \tag{56}$$
$$|1\rangle \mapsto |2\rangle \tag{57}$$
$$|2\rangle \mapsto |1\rangle \tag{58}$$
$$|3\rangle \mapsto |3\rangle. \tag{59}$$

The matrix representation of M is given by

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{60}$$

Generalising this to larger numbers of qubits stored is done via the same thought process. Taking the example of storing three qubits,

$$\text{Qubit 1 } a|0\rangle + b|1\rangle$$
$$\text{Qubit 2 } c|0\rangle + d|1\rangle$$
$$\text{Qubit 3 } e|0\rangle + f|1\rangle.$$

When stored in the order above, the qudit state is given by

$$ace\,|0\rangle + ade\,|1\rangle + bce\,|2\rangle + bde\,|3\rangle + ace\,|4\rangle + adf\,|5\rangle + bcf\,|6\rangle + bdf\,|7\rangle\,, \tag{61}$$

which again, when converted to binary numbers, can be expressed as the product state

$$(e\,|0\rangle + f\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \otimes (a\,|0\rangle + b\,|1\rangle)\,. \tag{62}$$

If we want to retrive qubit 1 then we require M to perform the following map

$$|1\rangle = |001\rangle \mapsto |100\rangle = |4\rangle \tag{63}$$
$$|3\rangle = |011\rangle \mapsto |110\rangle = |6\rangle \tag{64}$$
$$|4\rangle = |100\rangle \mapsto |001\rangle = |1\rangle \tag{65}$$
$$|6\rangle = |110\rangle \mapsto |011\rangle = |3\rangle \tag{66}$$

with all other basis states mapped to themselves as they are identical when switching the first and last digits of their binary representations. Having operated M on the qudit state, we can now run the circuit backwards in order to retrieve qubit 1. As with retrieving Bell pairs, any qubit in the state $|0\rangle$ can be used to retrieve qubit 1.

5 Conclusions

Hello

Acknowledgements

The author would like to thank...

References

[1] Charles H. Bennett and Stephen J. Wiesner. ‘Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states’. In: *Phys. Rev. Lett.* 69 (20 Nov. 1992), pp. 2881–2884. DOI: 10.1103/PhysRevLett.69.2881. URL: https://link.aps.org/doi/10.1103/PhysRevLett.69.2881.

[2] Charles H. Bennett and Gilles Brassard. ‘Quantum cryptography: Public key distribution and coin tossing’. In: vol. 560. Jan. 1984, pp. 175–179. DOI: 10.1016/j.tcs.2011.08.039.

[3] Charles H. Bennett et al. ‘Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels’. In: *Phys. Rev. Lett.* 70 (13 Mar. 1993), pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895.

[4] Neil Shenvi, Julia Kempe, and K. Birgitta Whaley. ‘Quantum random-walk search algorithm’. In: *Phys. Rev. A* 67 (5 May 2003), p. 052307. DOI: 10.1103/PhysRevA.67.052307. URL: https://link.aps.org/doi/10.1103/PhysRevA.67.052307.

[5] Andrew M. Childs. 'Universal Computation by Quantum Walk'. In: *Physical Review Letters* 102.18 (May 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.102.180501. URL: http://dx.doi.org/10.1103/PhysRevLett.102.180501.

[6] Viv Kendon. 'Decoherence in quantum walks – a review'. In: *Mathematical Structures in Computer Science* 17.06 (Nov. 2007). ISSN: 1469-8072. DOI: 10.1017/s0960129507006354. URL: http://dx.doi.org/10.1017/S0960129507006354.

[7] Andrew M. Childs et al. 'Exponential algorithmic speedup by a quantum walk'. In: *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC '03* (2003). DOI: 10.1145/780542.780552. URL: http://dx.doi.org/10.1145/780542.780552.

[8] Taira Giordani et al. 'Entanglement transfer, accumulation and retrieval via quantum-walk-based qubit-qudit dynamics'. In: *arXiv e-prints* (Oct. 2020). arXiv: 2010.07127 [quant-ph]. URL: https://arxiv.org/abs/2010.07127.

[9] J. Eisert et al. 'Classical Information and Distillable Entanglement'. In: *Physical Review Letters* 84.7 (Feb. 2000), pp. 1611–1614. ISSN: 1079-7114. DOI: 10.1103/physrevlett.84.1611. URL: http://dx.doi.org/10.1103/PhysRevLett.84.1611.

[10] G. Vidal and R. F. Werner. 'Computable measure of entanglement'. In: *Physical Review A* 65.3 (Feb. 2002). ISSN: 1094-1622. DOI: 10.1103/physreva.65.032314. URL: http://dx.doi.org/10.1103/PhysRevA.65.032314.

[11] Ben Tregenna et al. 'Controlling discrete quantum walks: coins and initial states'. In: *New Journal of Physics* 5 (July 2003), pp. 83–83. ISSN: 1367-2630. DOI: 10.1088/1367-2630/5/1/383. URL: http://dx.doi.org/10.1088/1367-2630/5/1/383.

Appendices

A Proof of No-Cloning Theorem

Here we present a proof of the no-cloning theorem.

Proof. Let $\mathcal{H}$ be a Hilbert space with two quantum systems A and B. Assume there exists a unitary operation $U \in \mathcal{H}$ that can clone any arbitrary $|\psi\rangle_A$ onto any arbitrary $|\phi\rangle_B$.

$$\begin{aligned} U\,|\psi\rangle \otimes |\phi\rangle &= |\psi\rangle \otimes |\psi\rangle && \text{(A.1)} \\ \implies U^{-1}\,|\psi\rangle \otimes |\psi\rangle &= |\psi\rangle \otimes |\phi\rangle && \text{(A.2)} \\ \implies U^{\dagger}\,|\psi\rangle \otimes |\psi\rangle &= |\psi\rangle \otimes |\phi\rangle && \text{(A.3)} \end{aligned}$$

Suppose we have any two arbitrary $|\psi\rangle_A, |\tilde{\psi}\rangle_A$ s.t. $0 \leq \langle\psi|\tilde{\psi}\rangle \leq 1$.

$$\begin{aligned} \langle\psi|\tilde{\psi}\rangle &= \langle\psi|\tilde{\psi}\rangle\,\langle\phi|\phi\rangle && \text{(A.4)} \\ &= (\langle\psi| \otimes \langle\phi|)(|\tilde{\psi}\rangle \otimes |\phi\rangle) && \text{(A.5)} \\ &= (\langle\psi| \otimes \langle\psi|)UU^{\dagger}(|\tilde{\psi}\rangle \otimes |\tilde{\psi}\rangle) && \text{(A.6)} \\ &= (\langle\psi| \otimes \langle\psi|)(|\tilde{\psi}\rangle \otimes |\tilde{\psi}\rangle) && \text{(A.7)} \\ &= \langle\psi|\tilde{\psi}\rangle^2 && \text{(A.8)} \\ \implies\ \langle\psi|\tilde{\psi}\rangle &= 0 \text{ or } \langle\psi|\tilde{\psi}\rangle = 1 && \text{(A.9)} \end{aligned}$$

which is a contradiction. □