Self-routing Quantum Sparse Crossbar Packet Concentrators

Journal:	*Transactions on Computers*
Manuscript ID:	TC-2008-12-0598.R1
Manuscript Type:	Regular
Keywords:	quantum computing, quantum information, quantum switching, packet switching, concentrator switches

Self-Routing Quantum Sparse Crossbar Packet Concentrators

Rahul Ratan, A. Yavuz Oruç
Department of Electrical and Computer Engineering,
University of Maryland, College Park, MD 20742

Abstract

Quantum switching networks are derived from conventional switching networks by replacing the classical switches by quantum switches. We give the quantum circuit design and routing of an $n \times m$ network, called a quantum concentrator that can direct quantum bit packets in arbitrary quantum states from any of its k inputs to some of its k outputs where $1 \leq k \leq m \leq n$. Our designs are based on sparse crossbars which are rectangular grids of 2×2 crosspoints. Sparse crossbar concentrator structures with theoretically minimum crosspoint complexities for any values of n and m are well-known. We transform two such families of optimal concentrators, called fat-slim and banded sparse crossbars into quantum networks and provide self-routing algorithms for these families of concentrators. In this process we extend the notion of packet concentration to a quantum network and design self-routing quantum crosspoints from quantum gates. We address issues critical to quantum operation like reversibility and localized self-routing and give a rigorous proof that quantum fat-slim and banded sparse crossbar concentrators are self-routable. The self-routing algorithms described in the paper can be used for both quantum and classical sparse crossbar concentrators by the linearity property of all quantum systems.

1 Introduction

Quantum computation and quantum information science are areas of research which have rapidly gained in prominence from their origins around three decades ago. The unique properties of quantum systems as manifested in the form of quantum parallelism and entanglement have been used in finding efficient solutions for classically intractable problems. The interest in this area received a major boost with the discovery of some seminal algorithms which have demonstrated that quantum systems can be used to solve interesting exponential complexity problems with speedups that are impossible in classical computing. Two well-known examples are Shor's polynomial time algorithm for finding prime factors of a composite number [1] and

Grover's search algorithm that can search an unstructured database with n elements in $O(\sqrt{n})$ time steps [2].

Any realistic scenario for the future scaling up of quantum computing involves spatially distributed quantum devices which interact with each other. Several recent schemes for large scale quantum computer architectures based on solid-state silicon have been proposed [3, 4] which are projected to provide the scalability required to achieve a useful computational substrate. The issue of quantum data transport has been recognized as a particularly critical requirement in upcoming silicon-based quantum computing technologies [5, 6]. Spatially distributed components introduce the need for *quantum wires* over which quantum data can be transmitted but building quantum wires and transferring quantum bits (qubits) is a non-trivial operation as, in general, quantum bits cannot be copied, which is a consequence of the no-cloning theorem [7]. Proposals for building quantum wires in solid state technologies include the quantum swapping and teleportation based architectures in [5, 6, 8]. The high cost of such wires implies that the $O(n^2)$ wires needed to interconnect n quantum devices can be a major bottleneck in implementing quantum systems.

This cost can be greatly reduced by using efficient switching network designs. The basic premise behind this idea is to use arrangements of reconfigurable switches with input quantum bits on their own quantum wires and then route them to the required destination. These switches are represented using quantum circuits composed of quantum gates. In addition to reducing wire count, reconfigurable quantum switches can form integral parts of the quantum data distribution system in envisioned architectures for scalable quantum computing. For example, in the Quantum Logic Array (QLA) microarchitecture proposed in [9], the high-level quantum computer structure consists of logical quantum bits connected with a programmable communication network in which integrated switch islands are used to redirect quantum data from nearby logical quantum bits.

The first design of a network for permuting individual quantum bits was given by Tsai and Kuo [10]. They gave a method to map a decomposition of any given permutation in terms of disjoint cycles and transpositions onto a quantum switch circuit which realizes that particular permutation. Since the circuit realizes only one permutation, it needs to be made anew for each new permutation to be realized. The first quantum switch network using reconfigurable switches to route groups of quantum bits was given by Shukla et. al. in [11, 12]. This quantum network can permute quantum information packets between its n inputs and n outputs. It was shown that

this network realizes $n^{n/2}$ permutations and can be used to resolve blocking when transmitting classical packets by creating a superposition of packets whenever they contend for the same wire in the network. Recently, Cheng and Wang [13] have proposed a quantum merge sorting-based network that can route all $n!$ permutations using $O(n \log^3 n)$ gates. Switching network configurations suitable for quantum networks have also been identified [14]. All these networks realize ordered connection maps, i.e., the particular output to which an input is connected is specified beforehand. In this paper we focus on the quantum circuit design and routing of concentrator switches that realize unordered connections between a set of inputs and a set of outputs. It is well-known that concentrators and other similar switches like extensive graphs and expanders are building blocks of more powerful interconnection networks such as nonblocking networks [15, 16].

Formally, an (n, m)-concentrator is a switching network with n inputs and m outputs, $1 \leq m \leq n$ in which any set of k inputs can be routed over nonintersecting paths to some k outputs, $1 \leq k \leq m$, but without the order specified. This is unlike other switches such as permutation networks which provide ordered connections between their inputs and outputs.

A family of concentrators, called *sparse crossbar concentrators*, can be designed using a grid or matrix of crosspoints with m rows and n columns where a crosspoint is placed between column i and row j if there exists an edge between input i and output j. Explicit sparse crossbar concentrator structures with theoretically minimum crosspoint complexities for any arbitrary values of n and m are well-known [17, 18]. Our main contribution in this paper is to transform two such families of optimal concentrators, called fat-slim and banded sparse crossbars into quantum networks and provide self-routing algorithms for these families of concentrators.

We first give an interpretation of concentration in a quantum network. In a classical concentrator network, packets for concentration are assigned at inputs and these input packet assignments can be issued only one assignment at a time. In a quantum concentrator, packets consist of quantum bits and thus represent a superposition of assignment patterns of packets which can be concentrated all at once by such a network due to the principle of quantum parallelism. This aspect is what distinguishes quantum concentrators from their classical counterparts. For example, consider a concentrator in which three inputs, say X, Y and Z, have packets which have to be concentrated. Suppose input X has two packets represented as x_1 and x_2, input Y has one packet y_1 and input Z has two packets z_1 and z_2. Y generates a "pure" packet while X and Z generate quantum packets by creating a superposition of both their respective packets,

and all three input sources then push their packets into a quantum concentrator. The outcome is that all the four possible input packet patterns: (x_1, y_1, z_1), (x_1, y_1, z_2), (x_2, y_1, z_1), (x_2, y_1, z_2) are routed in parallel and the output is a superposition of these four packet patterns each of which corresponds to the output obtained by concentrating one of the input packet patterns.

In the process of designing a quantum concentrator, we address some issues particular to quantum systems. One such issue is the reversibility constraint of quantum information processing. All quantum operations are inherently reversible in nature. This notion of reversibility is exactly the same as that commonly understood for any input-output mapping, i.e., given the output state of a quantum system, the corresponding input state can be uniquely determined. A "rectangular" structure like an (n, m)-sparse crossbar concentrator where the number of inputs, n, is not equal to the number of outputs, m, is inherently non-reversible. We devise a way to make (n, m)-crossbar concentrators "square" by using additional lines on the input and output sides of such concentrators and ensuring that valid packets for concentration are routed only among the original n inputs and m outputs.

Additionally, in a concentrator, a subset of inputs, say $\mathcal{I}_s$ can, in general, be matched to multiple subsets of outputs and a subset of outputs, say $\mathcal{O}_s$ can be the matching for multiple input subsets. Even when $\mathcal{O}_s$ is the only matching for $\mathcal{I}_s$, there may be multiple settings for the internal crosspoints which realize this matching. As a simple example, consider an (n, m)-concentrator in which a k-input subset and a k output subset are interconnected by a $k \times k$ full crossbar, $k \leq m$. Also assume that the k inputs in the k-input subset are not connected to any other outputs. Then obviously this k-input subset can be matched to only one output set of size k but all possible $k!$ one-to-one maps are possible.

Thus, to ensure reversibility a routing algorithm is needed to determine the crosspoint settings and fix the output matching subset for a given input subset. Even though efficient centralized routing algorithms for optimal crossbar concentrators with $O(\log n)$ delay for n tree-connected processors and $O(n \log n)$ delay for a single processor are known [18], [19], they cannot be adopted for quantum concentrators. It is critical to have a self-routing algorithm for quantum concentration in which the state of a crosspoint is determined by using only the local information from the incoming packet headers. An important property of self-routing packets is that the control quantum bits used to configure the crosspoint settings can be restored back to their original states easily thus preventing loss of information due to decoherence.

The rest of this paper is organized as follows. In Section 2, we introduce the basic quantum

circuit concepts that will be used in the design and routing of quantum concentrators. We also present a quantum packet concentrator model that will be used to describe our self-routing algorithms. In Section 3, we give a brief overview of classical sparse-crossbar concentrators. In Section 4, we define the functionality of quantum sparse-crossbar concentrators, present the design of quantum sparse crossbar concentrators and describe a self-routing algorithm for such concentrators. In Section 5, we prove the correctness of this algorithm for the optimal banded and fat-slim quantum crossbar concentrators. In Section 6, we give an example to describe the concentration process on a quantum sparse crossbar concentrator and address the issue of routing more than m packets on an (n, m)-quantum crossbar concentrator. In Section 7, we describe how to restore the state of auxiliary qubits to prevent decoherence and conclude the paper with the analysis of the gate count and routing delay, and remarks on remaining questions and future work in Section 8.

2 Quantum Circuits and Packets

In this section we give a brief description of basic concepts related to quantum information, quantum circuits, quantum gates, and quantum packets.

2.1 Qubits

The indivisible unit of classical information is the bit: an object that can take either one of two values: 0 or 1. The corresponding unit of quantum information is the quantum bit or *qubit*. Unlike a classical bit, a qubit can take values which are, in some sense, a combination of the values 0 and 1, i.e., it can be simultaneously be both 0 and 1. Formally, a qubit's state is represented as a unit vector in a two-dimensional complex Hilbert space and is expressed as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad |\alpha|^2 + |\beta|^2 = 1;\ \alpha, \beta \in \mathbb{C} \tag{1}$$

The vectors $|0\rangle$ and $|1\rangle$ constitute an orthonormal basis for this space. These two vectors are referred to as the computational basis vectors. We can perform a measurement that projects the state of this qubit onto the computational basis, i.e., the measurement projects $|\psi\rangle$ onto the basis $\{|0\rangle, |1\rangle\}$. The outcome of the measurement is not deterministic— the probability that we observe the result to be $|0\rangle$ is $|\alpha|^2$ and the probability that we observe the result to be $|1\rangle$ is $|\beta|^2$. α and β are referred to as the probability amplitudes of the states $|0\rangle$ and $|1\rangle$ respectively.

The state of an n-qubit system can be expressed as vector in a complex Hilbert space of dimension 2^n. This 2^n dimensional space is a tensor product of the n two-dimensional spaces

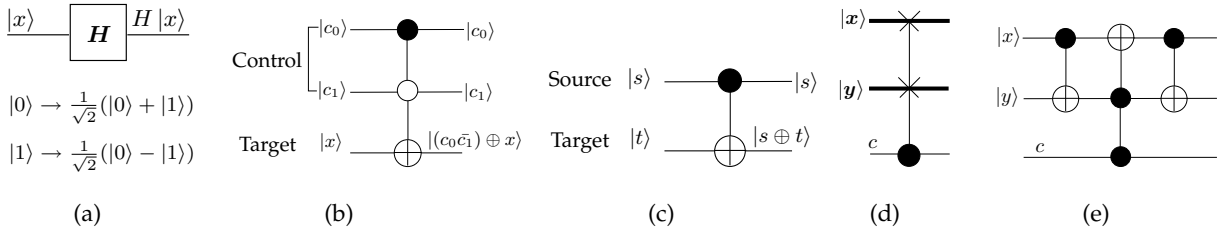


Fig. 1: Quantum gates: (a) Hadamard gate (b) Controlled-controlled not gate (c) Controlled-not (CNOT) gate (d) Controlled swap or switch gate (e) Switch gate made using CNOT gates.

representing individual qubits. The orthonormal basis for this space can be chosen as the states in which each qubit has a definite value, either $|0\rangle$ or $|1\rangle$. A general normalized vector representing an n-qubit state can be expanded in this basis as

$$\begin{aligned} |\psi\rangle &= \alpha_0 \,|00\cdots00\rangle + \alpha_1 \,|00\cdots01\rangle + \cdots + \alpha_{2^n-2} \,|11\cdots10\rangle + \alpha_{2^n-1} \,|11\cdots11\rangle \\ &= \sum_{i=0}^{2^n-1} \alpha_i \,|i\rangle \end{aligned} \tag{2}$$

where we have associated with each string the number that it represents in binary notation, ranging in value from 0 to $2^n - 1$. Here the α_i's are complex numbers satisfying $\sum_i |\alpha_i|^2 = 1$. A measurement of all n qubits by projection of each onto the $\{|0\rangle, |1\rangle\}$ basis, yields the outcome $|i\rangle$ with probability $|\alpha_i|^2$ [20].

2.2 Quantum Gates

The state of qubits is transformed using quantum gates and circuits composed of such gates. The quantum gate formalism was first proposed by Deutsch [21]. A quantum gate is a linear, unitary transformation on the space of qubit state vectors. The unitary nature of these transformations implies that quantum gates are *reversible*, i.e., given the state of qubits at the output of a gate, the input state can be uniquely determined. The unitarity also implies that the gates preserve the norm of the input state which amounts to preserving probability. These requirements of reversibility and norm preservation are basic axioms of quantum theory.

An example of a single qubit gate is the Hadamard gate, H, (Figure 1(a)) which transforms the basis vectors $|0\rangle$ and $|1\rangle$ as

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \;\; |1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{3}$$

In the above mapping we say that the basis vectors $|0\rangle$ and $|1\rangle$ are put in an equal superposition by the Hadamard gate, as after the transformation, the probability of observing either of the

basis vectors at the output is equal to 1/2. Thus, the Hadamard gate can be considered a quantum randomizer which takes a 0 or 1 bit and transforms it so that the output is either 0 or 1 with probability 1/2 [22].

A gate is completely specified by the mapping it performs on the basis vectors as all the rest of the input states can be represented as vectors which are a linear combination of these basis vectors. In the case of the Hadamard gate this means that an input qubit in the general state $\alpha|0\rangle + \beta|1\rangle$ would be transformed to $\frac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\beta}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$.

An n-bit quantum circuit can simultaneously act on all the 2^n components of the input state and transform them according to some specified mapping at once. This is the source of massive *quantum parallelism*. To make this more clear, suppose we are interested in finding the properties of a function $f(i)$ which takes the n-bit binary string i as input. The table of values for $f(i)$ is of size 2^n and is clearly infeasible to calculate for large n. But, with a quantum computer, U_f that acts according to

$$|i\rangle|0\rangle \xrightarrow{U_f} |i\rangle|f(i)\rangle \tag{4}$$

When we write any two qubit states side-by-side, it means we are taking a tensor product, thus $|i\rangle|0\rangle = |i\rangle \otimes |0\rangle$. We can put the input register consisting of the qubits corresponding to i in a superposed state similar to the one in Eqn. (2):

$$\underbrace{\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)}_{n\text{qubits}} = \frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle \tag{5}$$

where we have taken the tensor product of the n qubits to get the complete state. By computing $f(i)$ only once, we can generate a state

$$\frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle|0\rangle \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle|f(i)\rangle \tag{6}$$

All the global properties of f are encoded in this state and can be extracted if an efficient method is devised. This is the kind of *massive parallelism* Shor uses in his famous factorization algorithm [1]. This same parallelism enables us to probabilistically combine packets in quantum switching networks. The input to the quantum switching network can be a superposition of multiple packet assignments, all of which are routed in parallel to the outputs. The switching network operates in different switch configurations for different packet assignments. We give a more detailed explanation of this concept later in the paper for quantum concentrators.

Among gates which operate on multiple qubit inputs, the most common type of gates used in quantum circuits are the controlled quantum gates. A controlled gate's function is determined

by the state of some control qubits. For example, the controlled-controlled-not (CC-NOT) gate shown in Figure 1(b), with two control qubits c_0 and c_1 performs the following transformation

$$|c_0, c_1, x\rangle \xrightarrow{\boldsymbol{CC-NOT}} |c_0, c_1, (c_0.\bar{c_1} \oplus x)\rangle \tag{7}$$

thus, it inverts x when $c_0 = 1$ and $c_1 = 0$. We use the notational convention $|c_0, c_1, x\rangle = |c_0\rangle\,|c_1\rangle\,|x\rangle$. If a gate becomes active when a control qubit is 1, then that is indicated by a solid circle, (for c_0) and if a gate becomes active when a control qubit is 0 then that is indicated by an open circle, (for c_1). The CC-NOT gate transforms the basis vectors $|100\rangle$ and 101 to 101 and 100 respectively and leaves all the remaining six basis vectors unchanged. The qubit affected by the operation of a controlled gate is called the target qubit. If we initialize x to 0, then this gate can be viewed as a *quantum comparator* which sets the target qubit to $|1\rangle$ when $c_0 < c_1$ and leaves it unchanged otherwise.

2.3 Quantum Copy and Switch Gates

The simplest controlled gate is the controlled-not (CNOT) gate shown in Figure 1(c). The function of this gate is given by the mapping:

$$|s\rangle\,|t\rangle \xrightarrow{CNOT} |s\rangle\,|s \oplus t\rangle \tag{8}$$

Hence bit t is inverted when $s = 1$ and remains unchanged when $s = 0$. This gate functions as a NOT gate for the lower or target qubit when the control qubit is in state $|1\rangle$. When $t = 0$ we see that the mapping is of the form $|s\rangle\,|0\rangle \xrightarrow{CNOT} |s\rangle\,|s\rangle$, thus a CNOT gate can also be viewed as a copier which copies the upper or source qubit on to the lower or target qubit when the target qubit is initialized to state $|0\rangle$. Note that this copy operation can only be done when the upper qubit is in one of the two basis states: $|0\rangle$ or $|1\rangle$. For a source qubit in the general state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, the mapping is given by:

$$(\alpha\,|0\rangle + \beta\,|1\rangle)_s\,|0\rangle_t \xrightarrow{CNOT} \alpha\,|00\rangle_{st} + \beta\,|11\rangle_{st} \tag{9}$$

The output state is either 00 with probability $|\alpha|^2$ or 11 with probability $|\beta|^2$ and we have copied 0 and 1 bits to the target. We shall use such gates in our concentrator design.

The basic gate used to build quantum switching networks is the controlled swap gate or switch gate, which is shown in Figure 1(d). A switch gate is a multi-qubit gate which swaps two sets of qubits or quantum packets when a control qubit c is $|1\rangle$ and passes them through unchanged, otherwise [11, 12]. These two states of the switch gate are referred to as the cross

and through states respectively. Thus, this gate can be used as a reconfigurable 2×2 switch to route quantum bit packets. The function of this gate can be represented as

$$|\boldsymbol{x}\rangle |\boldsymbol{y}\rangle |0\rangle_c \xrightarrow[Through]{Switch} |\boldsymbol{x}\rangle |\boldsymbol{y}\rangle |0\rangle_c \tag{10}$$

$$|\boldsymbol{x}\rangle |\boldsymbol{y}\rangle |1\rangle_c \xrightarrow[Cross]{Switch} |\boldsymbol{y}\rangle |\boldsymbol{x}\rangle |1\rangle_c \tag{11}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are equal length binary strings. The thick lines in Figure 1(d) for $\boldsymbol{x}$ and $\boldsymbol{y}$ indicate that there are multiple qubits on them. An implementation of the switch gate with strings $\boldsymbol{x}$ and $\boldsymbol{y}$ of length 1, using two controlled-not (CNOT) and one CC-NOT gate is shown in Figure 1(e). If $\boldsymbol{x} = x_1x_0$ and $\boldsymbol{y} = y_1y_0$ are strings of length two, then the transformation done by the switch gate is given by:

$$|x_1, x_0\rangle |y_1, y_0\rangle |0\rangle_c \xrightarrow[Through]{Switch} |x_1, x_0\rangle |y_1, y_0\rangle |0\rangle_c \tag{12}$$

$$|x_1, x_0\rangle |y_1, y_0\rangle |1\rangle_c \xrightarrow[Cross]{Switch} |y_1y_0\rangle |x_1, x_0\rangle |1\rangle_c \tag{13}$$

2.4 Quantum Packets and Concentration Assignments

A quantum packet consists of a set of *data qubits* and one additional qubit which we refer to as the *routing qubit*. We assume that quantum packets composed of qubits are routed over a quantum concentrator. Reversibility considerations in quantum systems mean that unlike classical systems no connecting wire or input/output line can remain empty. We use the routing qubit to overcome this constraint, the routing qubit is set to $|1\rangle$ to indicate the presence of a quantum packet and to $|0\rangle$ to indicate an empty wire or absence of a packet. Note that in the rest of this chapter whenever we write $|a\rangle$, where a is a binary variable, then this represents the state of a quantum bit. The binary variable itself is indicated as a. A quantum packet is defined as [23]:

Definition 1 (Quantum packet)**.** Let an input have m n_d-bit packets, $d_1, \ldots, d_m$. If packet d_i is to be concentrated with probability p_i, then the source at this input feeds into the concentrator a *quantum packet* of the following form:

$$\sum_{i=1}^{m} \alpha_i |r_i, d_i\rangle \tag{14}$$

where $|\alpha_i|^2 = p_i$ and $r_i = 1$. We refer to the individual components of the quantum packet, the bit strings (r_i, d_i) as *classical packets*. The routing bits in the classical packets are r_i. The length of the quantum packet is $n_d + 1$ qubits. We refer to these strings as classical packets as

they represent a basis state (with no superposition) of the constituent qubits and any group of quantum bits in a basis state are conceptually equivalent to a group of classical bits having the same value.

If the input source has no packets to concentrate then the empty line is indicated by a single $n_d + 1$-bit string in which the routing bit is set to 0, and the data bits can be set to any arbitrary value.

An input *concentration pattern* for an n-input concentrator is a sequence of classical packets, each of which belongs to a quantum packet on the n inputs from top to bottom. A *quantum concentration assignment* is a superposition of a set of concentration patterns. We define these terms formally as follows:

Definition 2 (Quantum concentration assignment)**.** A *quantum concentration assignment* for an n-input concentrator is a superposition of a set of t concentration patterns of the form:

$$\sum_{j=1}^{t} \gamma_j \left|(r_{j,1}, d_{j,1}), \cdots, (r_{j,n}, d_{j,n})\right\rangle \tag{15}$$

where the *concentration pattern* $|(r_{j,1}, d_{j,1}), \cdots, (r_{j,n}, d_{j,n})\rangle$ consists of n classical packets in which $(r_{j,i}, d_{j,i})$ is the jth classical packet at input i. $|\gamma_j|^2$ is the probability of the jth concentration pattern being realized with $\sum_{j=1}^{t} |\gamma_j|^2 = 1$.

If the quantum packets at all the inputs are independent and of the kind given in Eqn. (14) then the quantum assignment can be expressed as a tensor product of the quantum packets as follows:

$$\bigotimes_{i=1}^{n} |Q_i\rangle = \bigotimes_{i=1}^{n} \left(\sum_{j=1}^{t_i} \alpha_{ij} \left|r_{ij}, d_{ij}\right\rangle \right) \tag{16}$$

where $|Q_i\rangle = \sum_{j=1}^{t_i} \alpha_{ij} \left|r_{ij}, d_{ij}\right\rangle$ is the quantum packet on input i. The tensor product in Eqn. (16) expanded to a quantum concentration assignment of the form in Eqn. (15) contains $\prod_{i=1}^{n} t_i$ concentration patterns.

As an example, consider three inputs indexed by $1, 2$ and 3 having the quantum packets: $\frac{1}{\sqrt{2}}(|1, d_{11}\rangle + |1, d_{12}\rangle)$, $|1, d_{21}\rangle$ and $\frac{1}{\sqrt{3}} |1, d_{31}\rangle + \frac{\sqrt{2}}{\sqrt{3}} |1, d_{32}\rangle$ respectively. Then the quantum concentration assignment is given by:

$$\begin{aligned} & \frac{1}{\sqrt{2}} (|1, d_{11}\rangle + |1, d_{12}\rangle) \otimes |1, d_{21}\rangle \otimes \left(\frac{1}{\sqrt{3}} |1, d_{31}\rangle + \frac{\sqrt{2}}{\sqrt{3}} |1, d_{32}\rangle \right) \\ = \; & \frac{1}{\sqrt{6}} |(1, d_{11}), (1, d_{21}), (1, d_{31})\rangle + \frac{1}{\sqrt{3}} |(1, d_{11}), (1, d_{21}), (1, d_{32})\rangle \end{aligned}$$

$$+\frac{1}{\sqrt{6}}\,|(1,d_{12}),(1,d_{21}),(1,d_{31})\rangle + \frac{1}{\sqrt{3}}\,|(1,d_{12}),(1,d_{21}),(1,d_{32})\rangle \tag{17}$$

Thus the quantum concentration assignment consists of a superposition of four concentration patterns, two of which have a probability $1/3$ and the other two a probability $1/6$ of being observed on measurement. A quantum concentrator can route such patterns contained in the input quantum concentration assignment in parallel.

3 Classical Sparse Crossbar Concentrators

An (n,m)-sparse crossbar network is a matrix of crosspoints or switches with m rows and n columns. Each crosspoint acts as a simple 2×2 switch which can either swap the data on its two inputs onto its outputs or pass them through unchanged. We refer to these two states of the crosspoint as the "cross" state and the "through" state respectively.

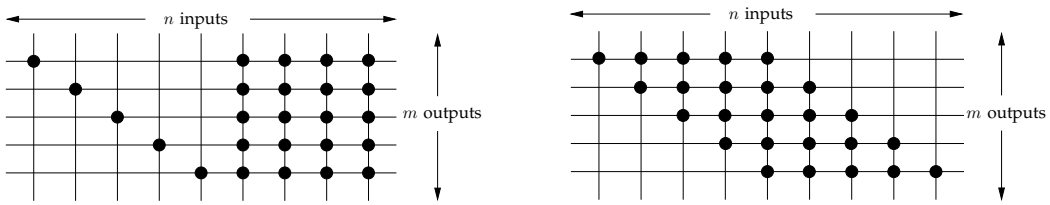
An (n,m)-sparse crossbar concentrator is a (n,m)-sparse crossbar in which any k inputs, $k \leq m$, can be routed over nonintersecting paths to some k outputs. Any sparse crossbar is a concentrator if its crosspoint distribution is such that the constraints of Hall's theorem are satisfied. This theorem is stated below:

Theorem 1 (Hall's Theorem)**.** *Let O be a finite set and let $Y_1, Y_2, \ldots, Y_r$ be arbitrary subsets of O. There exist distinct elements $y_i \in Y_i$, $1 \leq i \leq n$ if and only if the union of any k of $Y_1, Y_2, \ldots, Y_r$ contains at least k elements.*

Let the set O in the theorem denote the set of outputs of a sparse crossbar and $Y_1, Y_2, \ldots, Y_r$ represent the neighbor sets of some r inputs $s_1, s_2, \ldots, s_r$ respectively, i.e., Y_i is the subset of outputs in O to which input s_i can be connected, $1 \leq i \leq r$. Then if the union of $Y_1, Y_2, \ldots, Y_r$ contains at least r outputs for any choices of $s_1, s_2, \ldots, s_r$ in the input set, and any r, $1 \leq r \leq m$, then Hall's theorem implies that the sparse crossbar is a concentrator.

Nakamura and Masson in [24] gave a lower bound of $m(n-m+1)$ on the crosspoint complexity, i.e., number of crosspoints, for (n,m)-sparse crossbar concentrators by showing that each output needs to share crosspoints with at least $n-m+1$ inputs. Oruç et al in [17] and [18] gave explicit crossbar structures of optimal concentrators which achieved this bound for any n and m. They used Hall's theorem to show that certain $n \times m$ sparse crossbar structures with exactly $m(n-m+1)$ crosspoints can act as concentrators.

Two such optimal concentrators, the fat-slim and banded sparse crossbar concentrators, which we shall be using extensively, are shown in Figure 2. As seen in Figure 2(a), in a fat-slim crossbar



(a) Fat-slim crossbar concentrator. (b) Banded crossbar concentrator.

Fig. 2: Classical sparse crossbar concentrators, n=9, m=5.

concentrator, the input columns can be divided into a fat portion in which an input is connected to all outputs and a slim portion in which an input is connected to only one output with all slim inputs being connected to different outputs. In a banded crossbar concentrator (Figure 2(b)) all the crosspoints form a transverse band in the middle. Note that in these sparse crossbars each of the m outputs is connected to $n - m + 1$ inputs.

4 Quantum Sparse Crossbar Concentrators

An input concentration pattern for a concentrator is said to be *capacity achieving* if no greater than m packets in the pattern have their routing bits equal to 1, where m is the number of outputs of the concentrator. We call a quantum concentration assignment *capacity achieving* if all of its concentration patterns are capacity achieving. Also, we refer to packets with routing bit set to 1 as valid packets.

A quantum sparse crossbar network is obtained from a classical crossbar network by replacing the classical crosspoints by quantum crosspoints which can switch quantum packets. A quantum crosspoint can be viewed as a configurable 2×2 switch which either swaps or passes through unchanged to its outputs the two quantum packets incident at its inputs. Reversibility in quantum systems implies, that for a quantum sparse crossbar, unlike a classical sparse crossbar, each crosspoint needs to have qubits coming in on each of its two inputs and qubits leaving on each of its outputs. As mentioned earlier in Section 2.4 an absence of a packet or an empty wire is indicated by a quantum bit string with the routing bit set to 0. Thus, in the quantum domain, for an $n \times m$ sparse crossbar network, the m empty wires coming in from the left can be represented by blocks of quantum bits in which the routing bit is set to 0. We can imagine m additional packet sources at these wires which generate quantum bit blocks in which the routing bit is always set to 0. The same reasoning can be applied to the n empty wires leaving the sparse crossbar at the bottom. This means that they can be viewed as n additional outputs

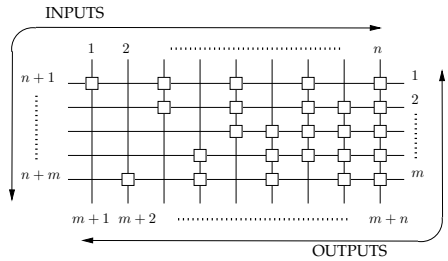


Fig. 3: Numbering of inputs and outputs in an $n \times m$ quantum sparse crossbar network.

and if the sparse crossbar is a concentrator then the exiting quantum bit strings on these outputs have their routing bits equal to 0 as long as the input pattern is capacity achieving. This in effect creates a "square" $(n+m) \times (n+m)$ crossbar network from a "rectangular" $n \times m$ network in which only the n inputs (on the top) can get valid packets. If the sparse crossbar is a concentrator and the input pattern is capacity achieving then all the valid packets are concentrated to the m outputs on the right hand side. If we number the inputs on the top $1, \ldots, n$ from left to right, the inputs on the left $n+1, \ldots, n+m$ from top to bottom, the outputs on the right $1, \ldots, m$ and outputs on the bottom $m+1, \ldots, m+n$ as shown in Figure 3 then:

Definition 3 (QSC(n, m))**.** An $n \times m$ quantum sparse crossbar network ($n \geq m$) is called an (n, m)-*quantum sparse crossbar concentrator* or QSC(n, m) if, any capacity achieving input pattern with k valid packets, where $k \leq m$, is routed such that these packets appear on some k of the first m outputs. That is, for a capacity achieving input concentration pattern $|P\rangle = |(r_1, d_1) \cdots (r_n, d_n)\rangle$, a pattern $|(0, d_{n+1}) \cdots (0, d_{n+m})\rangle$ of m packets each with routing qubit set to $|0\rangle$ and a set of auxiliary qubits initialized to state $|0\rangle$ the following transformation occurs:

$$|\underbrace{(r_1, d_1) \cdots (r_n, d_n)}_{|P\rangle\text{: from top}}\rangle|\underbrace{(0, d_{n+1}) \cdots (0, d_{n+m})}_{\text{from left}}\rangle\, |00 \cdots 0\rangle_{aux} \xrightarrow{QSC(n,m)}$$
$$|\underbrace{(r'_1, d'_1) \cdots (r'_m, d'_m)}_{\text{on right}}\rangle|\underbrace{(0, d'_{m+1}) \cdots (0, d'_{n+m})}_{\text{bottom}}\rangle\, |\Psi_P\rangle_{aux} \quad (18)$$

where if $\mathcal{R}_1 = \{|r_i, d_i\rangle \ \forall \ r_i = 1\}, \ 1 \leq i \leq n$, is the set of valid input packets and $\mathcal{R}_2 = \{|r'_j, d'_j\rangle \ \forall \ r'_j = 1\}, \ 1 \leq j \leq m$, is the set of valid packets at the outputs then $\mathcal{R}_2 = \mathcal{R}_1$. Here $|00 \cdots 0\rangle_{aux}$ represents the state of a set of auxiliary qubits all of which are in state $|0\rangle$.

In a crossbar network, a subset of inputs can potentially be matched to more than one set of outputs. Moreover, even if an input set can be matched to only one output set, it is possible that several one-to-one input-output maps within these sets realize the matching. Therefore,

some form of a routing algorithm is needed to fix the matched output set and the input-output mapping for a capacity achieving input pattern. If a quantum crosspoint is configured by using only the information contained in the quantum packets incident on its inputs, then we call such a quantum crosspoint as a self-routing crosspoint. A quantum sparse crossbar concentrator built using such crosspoints is called a *self-routing quantum sparse crossbar concentrator*. Hence, a self-routing QSC(n, m) is a sparse crossbar network built using self-routing quantum crosspoints which realizes all maps of the kind given in Eqn. (18).

We describe such an algorithm in the next section. The auxiliary qubits ensure reversible operation as their final state, $|\Psi_P\rangle$, encodes the state of the crossbar network, i.e., the states of the internal crosspoints or equivalently the input-output mapping.

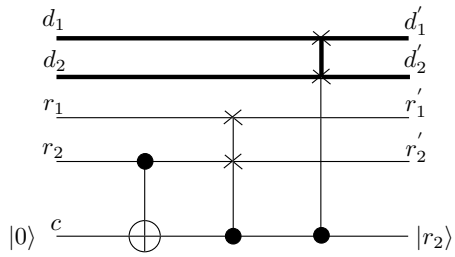
4.1 Self-Routing Quantum Crosspoints

In this section we give the design of self-routing quantum crosspoints and a routing scheme for sparse crossbars composed of such crosspoints.

A self-routing $n \times m$ quantum sparse crossbar is derived from a classical sparse crossbar structure as follows: The crosspoints in the classical sparse crossbar are replaced by quantum crosspoints, the circuit for which is given in Figure 4. Here the upper input with packet $|r_1,\ d_1\rangle$ corresponds to the input line incident on a crosspoint from the top and the lower input with packet $|r_2,\ d_2\rangle$ corresponds to the input line coming in from the left. The data bit strings d_1, d_2 and the routing bits r_1, r_2 are shown separately for clarity. Each crosspoint uses an auxiliary control qubit initialized to state $|0\rangle$, which is then set according to the map in Figure 5 and used to control the setting ("through" or "cross") of the switch gate. When the control qubit is set to state $|1\rangle$, the input packets get swapped and when the control qubit is in state $|0\rangle$, the input packets go through unchanged. In the crosspoint circuit the CNOT gate functions as a copier which sets the state of the control qubit to $|r_2\rangle$. This qubit is then used to control the two swap gates which switch r_1, r_2 and d_1, d_2 respectively. Thus, the input-output mapping performed by the quantum crosspoint can be represented as:

$$\begin{aligned} |r_1, d_1, 0, d_2\rangle\,|0\rangle &\xrightarrow[Through]{Crosspoint} |r_1, d_1, 0, d_2\rangle\,|0\rangle \\ |r_1, d_1, 1, d_2\rangle\,|0\rangle &\xrightarrow[Cross]{Crosspoint} |1, d_2, r_1, d_1\rangle\,|1\rangle \end{aligned} \tag{19}$$

We can see that the auxiliary control qubit is always set to state $|r_2\rangle$, i.e., the packets are swapped when the routing qubit of the packet coming in from the left (r_2) is in state $|1\rangle$ and go through



Inputs		Outputs				
r_2	r_1	r_2'	r_1'	d_2'	d_1'	State
0	0	0	0	d_2	d_1	Through
0	1	0	1	d_2	d_1	Through
1	0	0	1	d_1	d_2	Cross
1	1	1	1	d_1	d_2	Cross

Fig. 4: Circuit for the quantum crosspoint. Fig. 5: Input-output map for a quantum crosspoint.

unchanged when this qubit is in state $|0\rangle$. Although the switch control is determined fully by just r_2, we cannot use its corresponding qubit by itself (without the auxiliary control qubit) to set the switch. This is because the qubit for r_2 is a part of the packet incident on the lower input and itself needs to be switched along with that packet, so although this qubit can act as a control qubit for all the rest of the qubits in the packet, it cannot be a control qubit for switching itself. Also, we are not using r_1 in the switch to determine the routing state, but this bit is still relevant as eventually at the output of the concentrator valid packets are identified by examining the routing bit values. Another reason to consider r_1 is the fact that crosspoint switches are interconnected to form the crossbar, and the packet from the upper input at one crosspoint switch may be incident at the lower input of a switch in later stages. In this scenario the qubit corresponding to r_1 would be the auxiliary control for this later stage switch. As the state of the quantum crosspoint is determined fully by just using the information about the routing bits from the two input packets, the paths in the sparse crossbar are determined in a self-routing fashion.

4.2 The Self-Routing Scheme

Starting from input 1, the routing of packets proceeds from top to bottom in a column for an input and then to next higher numbered input in the next column. The control qubit is not restored to its original state immediately, we give a method to restore the control qubit at the output of the concentrator in Section 7.

Note that unlike in the classical case, all n inputs (plus the m additional inputs from the left) have quantum bit strings incident on them. We distinguish the subset of inputs having packets for concentration by the setting the routing qubits in the headers of packets at these inputs to 1. A self-routing quantum concentrator derived from a classical fat-slim $(5, 3)$-sparse crossbar concentrator is given in Figure 6. The square boxes in Figure 6 indicated by letters A–I are quantum crosspoints. In this concentrator valid packets (packets with routing bit 1) come only

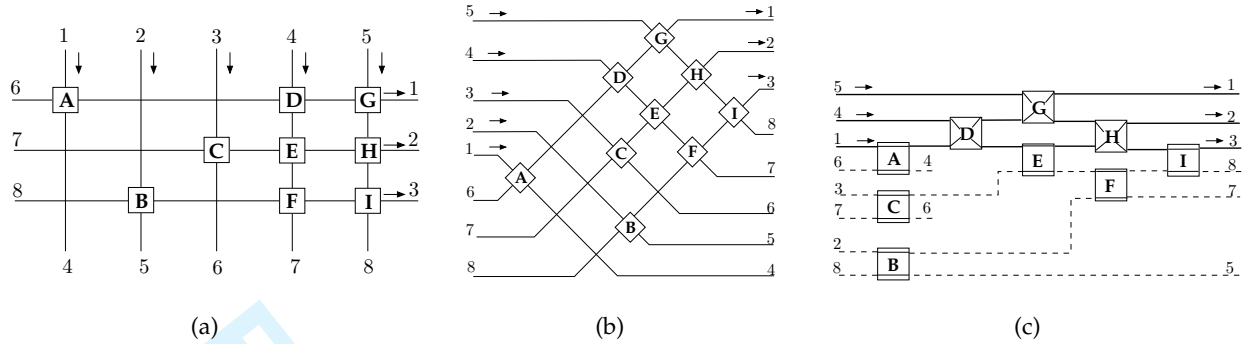


Fig. 6: Self-routing fat-slim QSC(5,3): (a) Crossbar structure (b) Crosspoint sequence in routing (c) Example for self-routing: Inputs 1, 4 and 5 are concentrated.

on inputs 1–5, and if the input pattern is capacity achieving, they exit only on outputs 1–3. These inputs and outputs are indicated by arrows in the figure. In the process of self-routing, the crosspoints are traversed from top to bottom in a column and from left to right in a row. The crossbar in Figure 6(a) is redrawn in Figure 6(b) to show the sequence in which the crosspoints are traversed during routing. Any sparse crossbar can be redrawn in this way if we rotate it counter-clockwise to make the diagonal vertical and write down the sequence of crosspoints encountered as we traverse from left to right. As a result of this redrawing we can see that all crosspoints lying on the diagonals which are oriented from top right to bottom left in the original crossbar form one vertical stage in the rotated version, e.g., G, E and H, F which lie on a diagonal in Figure 6(a) form independent stages in the crossbar in Figure 6(b). Note that D, C and B also lie on a single diagonal but since B and C are disjoint with each other and with D, they can be shifted from the second stage to the first, as shown in Figure 6(c). Figure 6(c) is Figure 6(b) with the inputs and outputs reordered to clearly show the planar and multi-stage structure of the sparse crossbar.

We now give an example to elucidate the self-routing process. Consider an input concentration pattern with valid packets at inputs 1, 4 and 5 for the crossbar shown in Figure 6(c), i.e., only packets appearing at inputs 1, 4 and 5 have routing bits set to 1. At all the other inputs the incident packets have routing bits set to 0. At crosspoint A the upper input, i.e., input 4 has a valid packet and the lower input has a packet with routing bit set to 0. Thus, this situation corresponds to the $r_1 = 1, r_2 = 0$ case in Figure 5 and A is set to pass the packets through unchanged. Proceeding to the next stage, we see that at crosspoint D, the routing bit of the packets at both the lower and upper inputs is 1, hence D swaps its input packets onto its

outputs. Continuing in this way we see that the packet from input 1 takes the path A→D→G to output 1. Similarly the packets from input 4 and input 5 take the paths D→E→H and G→H→I to outputs 2 and 3 respectively. The switch settings and the packet routes are shown in Figure 6(c). The routes taken by the valid packets is shown by solid lines while the routes taken by packets with routing bit equal to 0 are shown by dashed lines. Note that the path lengths between all the inputs and outputs are not equal, hence we need to introduce appropriate delay to make them all equal, accordingly, all output lines are extended till the end. Output lines 4 and 6 are not extended just to maintain clarity in the drawing but may be considered to extend till the last output stage.

The intuition behind the routing scheme is as follows: In the rotated crossbar as shown in Figure 6(b) the upper and lower input lines to a switch correspond to the lines coming in to the same switch from the top and left respectively in the original unrotated version. Similarly the upper and lower outputs in the rotated switches correspond to the line going out to the right and bottom respectively in the original unrotated switches. Hence, giving routing priority to the packet at the lower input in a quantum crosspoint is equivalent to routing according to the packet coming in from the left in the unrotated crossbar. If the routing bit of the lower packet is 1 then switch is set in a cross state, this is equivalent to passing a valid packet coming in from the left to the right irrespective of the packet incident from the top. Thus, once a valid packet is routed from the top to the right, it goes through unimpeded to the end of the row, in other words, once an input is matched to an output this decision remains unchanged for the subsequent duration of the routing process. The crosspoint is set in a through state when the packet at the lower input in the rotated switch has its routing bit equal to 0, this corresponds to a turn from left to the bottom in the unrotated crossbar. This observation combined with the previous argument means that in a column of the unrotated crossbar a packet gets passed from top to bottom from one row to the next until it encounters a crosspoint where the packet from the left has a routing bit set to 0.

This self-routing scheme can be used for any sparse crossbar structure, it is not limited to concentrators. An interesting question to ask is whether all known optimal sparse crossbar concentrator structures allow concentration when a self-routing scheme of the form described above is used to route packets. In the subsequent sections we prove that while this is not true for all optimal sparse crossbar concentrators, it holds for fat-slim and banded crossbars.

5 Self-Routing on Quantum Sparse Crossbar Concentrators

We prove that the self-routing scheme described in Section 4.1 concentrates packets up to the output capacity of the concentrator, for two families of sparse crossbar structures, namely the fat-slim and banded crossbars. All the proofs are given for a quantum input assignment consisting of a single concentration pattern. This is sufficient as the linearity of quantum systems implies that, for any general input quantum concentration assignment, these results apply to every concentration pattern contained in its superposition and hence to the entire input assignment. We first introduce some notation that will be used throughout the rest of the paper.

5.1 Notation

For a quantum (n, m)-sparse crossbar network ($n \geq m$) in which packets are routed using our self-routing scheme, we use the following notation:

1) The set of inputs is denoted by $\mathcal{I} = \{1, 2, \ldots, n\}$ and the output set by $\mathcal{O} = \{1, 2, \ldots, m\}$ where $1 \leq m \leq n$.
2) The $n \times m$ adjacency matrix, $\boldsymbol{A}$, is given by:

$$\boldsymbol{A} = \{a_{ij}\} = \begin{cases} 0, & \text{no crosspoint between input } i \text{ and output } j, \\ 1, & \text{crosspoint between input } i \text{ and output } j. \end{cases}$$

3) $\mathcal{A}_i = \{j : \text{for all } a_{ij} = 1, 1 \leq i \leq n\}$, is the neighbor set for input i, i.e., the set of outputs which can be connected to input i.
4) $\mathcal{X}_r = \{x_1, x_2, \ldots, x_r\}$ is an ordered set of r distinct inputs, i.e., $x_i \in \mathcal{I},\ 1 \leq i \leq r$, and $x_1 < x_2 < \cdots < x_r$ for all $r = 1, \ldots, m$.
5) $\mathcal{Y}_r = \{y_1, y_2, \ldots, y_r\}$ is the set of outputs to which the inputs in $\mathcal{X}_r$ are matched using self routing with output y_i being matched to input x_i, where, $y_i \in \mathcal{A}_{x_i}$, $i = 1, \ldots, r$. If input x_i can not be matched to any output then $y_i = \emptyset$, the empty set.
6) We denote subsets of $\mathcal{Y}_r$ as follows: $\mathcal{Y}_0 = \emptyset$, the empty set, $\mathcal{Y}_b = \{y_1, y_2, \ldots, y_b\}$, and $\mathcal{Y}_a^b = \{y_a, y_{a+1}, \ldots, y_b\},\ \forall\ a \leq b \leq r$.

We now show that using the self-routing scheme described above in Section 4.2, certain families of sparse crossbar concentrators, namely fat-slim and banded crossbar concentrators can correctly route any capacity achieving input pattern.

Lemma 1. *Let $\mathcal{Z} \subseteq \mathcal{O}$ be the subset of outputs to which packets have already been matched, using self-routing, before routing begins on input x_i, $1 \leq i \leq r$. Then, the output y_i, to which input x_i is*

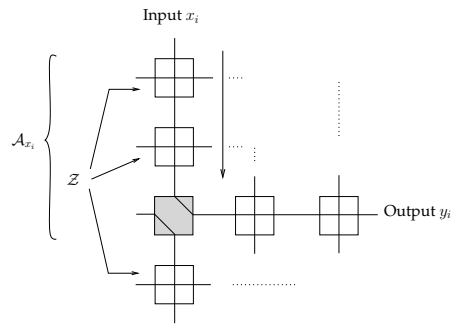


Fig. 7: Output matching y_i for input x_i.

matched is given by $y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap \mathcal{Z}'\}$, *where* $\mathcal{Z}' = \mathcal{O} \setminus \mathcal{Z}$ *is the complement of the set* $\mathcal{Z}$.

Proof: All outputs which have been concentrated to (matched) before routing the packet on input x_i, (i.e., in $\mathcal{Z}$) correspond to rows which have a packet with routing bit $r = 1$ coming from the left on them. We know from the table for the auxiliary control qubit (Figure 5) that all crosspoints in rows corresponding to set $\mathcal{Z}$ will be set to the "cross" state, see Figure 7. This means that only a crosspoint from the set of rows corresponding to $\mathcal{Z}'$ will be set to the "through" state. This in turn means $y_i \in \mathcal{Z}'$. Also, obviously, $y_i \in \mathcal{A}_{x_i}$. Thus, $y_i \in \mathcal{A}_{x_i} \cap \mathcal{Z}'$. Since the routing process proceeds from the top to bottom, i.e., in increasing order of row/output number, the lowest numbered available output is matched. Hence, $y_i = \min_z\{z \in \mathcal{A}_{x_i} \cap \mathcal{Z}'\}$. □

Lemma 2. *Let* y_i *be the output to which input* x_i *is matched,* $1 \leq i \leq r$. *Then,*

$$y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_{i-1})'\}$$

Proof: As the routing of packets at inputs is initiated in increasing order of input index from left to right, the packets at inputs $x_1, \ldots, x_{i-1}$ are routed before routing starts on input x_i. Thus, the set of outputs matched before routing starts on input x_i is $\mathcal{Y}_{i-1}$. Substituting $\mathcal{Z} = \mathcal{Y}_{i-1}$ in Lemma 1 we get the desired result. □

Lemma 2 essentially asserts that, at any input, the first or lowest numbered unmatched output is selected as a match for that input during self-routing. If an input x_i cannot be matched to any output in its neighbor set $\mathcal{A}_{x_i}$, then $y_i = \emptyset$.

Lemma 3. *If* $i \neq j$ *and* $y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_{i-1})'\} \neq \emptyset$ *and* $y_j = \min_z \{z \in \mathcal{A}_{x_j} \cap (\mathcal{Y}_{j-1})'\} \neq \emptyset$. *Then,* $y_i \neq y_j$, $1 \leq i, j \leq r \leq m$.

Proof: Without loss of generality, assume $i < j$, then $y_i \in \{y_1, \ldots, y_{j-1}\} = \mathcal{Y}_{j-1}$. Also $y_j = \min_z\{z \in \mathcal{A}_{x_j} \cap (\mathcal{Y}_{j-1})'\}$ implies that $y_j \in (\mathcal{Y}_{j-1})'$. Hence, $y_i \neq y_j$. □

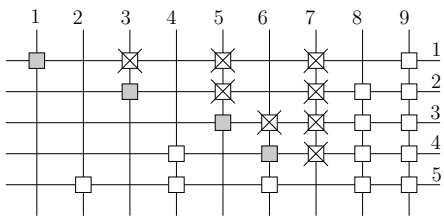


Fig. 8: Conflict in self-routing.

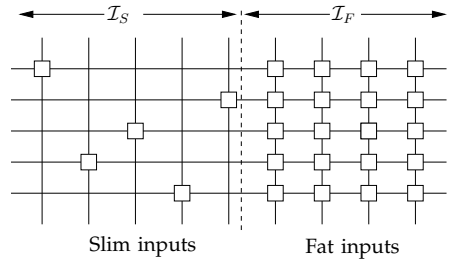


Fig. 9: Partitions of fat-slim crossbar.

Lemma 3 implies that the non-empty elements of $\mathcal{Y} = \{y_1, \ldots, y_r\}$ are all distinct. Thus, if all the elements of $\mathcal{Y}$ exist, i.e., are non-empty, then $\mathcal{Y}$ forms a matching for the inputs in the set $\mathcal{X} = \{x_1, \ldots, x_r\}$. Moreover, if such a matching exists for every r-input subset of a (n, m)-sparse crossbar network, $r \leq m$, then the network is an (n, m)-concentrator. From Lemma 2 the matching $\mathcal{Y}$ corresponds to the outputs chosen by self-routing, therefore, the concentrator thus obtained is self-routable. The theorem below follows:

Theorem 2. *An (n, m)-sparse crossbar network is a self-routing concentrator if*

$$y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_{i-1})'\} \neq \emptyset$$

for all $i = 1, \ldots, r$, and for every r-input ordered subset $\mathcal{X}_r = \{x_1, \ldots, x_r\}$, $1 \leq r \leq m$.

A quantum concentrator derived by replacing classical crosspoints by quantum crosspoints in a classical sparse crossbar concentrator is not always self-routable by the algorithm described above. One such scenario is shown in Figure 8. For the sparse crossbar in this figure the union of any k columns (or equivalently input neighbor sets) contains crosspoints in least k distinct rows (or outputs) for all $k = 1, \ldots, 5$, thus by Hall's theorem it is a $(9, 5)$-concentrator. The set of inputs with packets to be concentrated is given by $\mathcal{X}_5 = \{1, 3, 5, 6, 7\}$. The output to which an input is matched by following the self-routing scheme is indicated by the shaded crosspoints, therefore, input 1 is matched to output 1, input 3 to output 2 and so on. The crossed out crosspoints indicate the outputs to which an input is not matched during self-routing. The matched outputs for the first four inputs in $\mathcal{X}_5$ form the set $\mathcal{Y}_4 = \{1, 2, 3, 4\}$. Thus for input 7 (which is the fifth input in $\mathcal{X}_5$) we get $y_5 = \min_z \{z \in \mathcal{A}_7 \cap (\mathcal{Y}_4)'\} = \min_z \{z \in \{1, 2, 3, 4\} \cap \{5\}\} = \emptyset$. Hence, this concentrator can not self-route all input subsets using the algorithm we described.

We now show that fat-slim and banded sparse crossbar concentrators can self-route by showing that for these concentrators Theorem 2 is always satisfied. As part of the proofs we give explicit expressions for the input-output mapping realized for concentration on these structures.

5.2 Self-Routing Fat-Slim QSC(n, m)

Definition 4 (Fat-Slim Crossbar)**.** An (n, m)-sparse crossbar network is called fat-slim if we can partition the input set $\mathcal{I}$ into two subsets: $\mathcal{I}_S$ (slim inputs) and $\mathcal{I}_F$ (fat inputs) as shown in Figure 9 with neighbor sets for input i described as follows:

$$i \in \mathcal{I}_S \quad \Leftrightarrow \quad 1 \leq i \leq m;\ \mathcal{A}_i = \pi(i) \tag{20a}$$

$$i \in \mathcal{I}_F \quad \Leftrightarrow \quad m < i \leq n;\ \mathcal{A}_i = \{1, \ldots, m\} = \mathcal{O} \tag{20b}$$

where π is a permutation on the elements of the set $\{1, \ldots, m\}$.

Every fat-slim n, m-sparse crossbar is an optimal (n, m)-concentrator with $m(n - m + 1)$ crosspoints [17]. We now show that any capacity achieving input pattern can be self-routed on a fat-slim sparse crossbar concentrator.

Theorem 3. *For the fat-slim QSC(n, m) let $\mathcal{X} = \{x_1, x_2, \ldots, x_r\}$ be any ordered r-input subset where $x_1 < x_2 < \cdots < x_r$, $\forall r, 1 \leq r \leq m$. Then, there exists an output matching, $\mathcal{Y} = \{y_1, y_2, \ldots, y_r\}$ for $\mathcal{X}$ obtained as result of self-routing the fat-slim QSC(n, m) and it is given by*

$$y_i = \begin{cases} \pi(x_i), & x_i \in \mathcal{I}_S, \\ b_{i-(m-a)}, & x_i \in \mathcal{I}_F. \end{cases}$$

where $\mathcal{B} = \{b_1, \ldots, b_a\} = (\{\pi(x_i) \in \mathcal{I}_S\})'$ such that $b_1 < b_2 < \cdots < b_a$, $a = |\mathcal{B}|$, $i = 1, 2, \ldots, r$, .

Proof: In the ordered r-input set $\mathcal{X}$, let the first k ($k \leq r$) inputs belong to the slim section and the rest to the fat section.

If $x_1 \in \mathcal{I}_S$, we get

$$\begin{aligned} y_1 &= \min_z \{z \in \mathcal{A}_{x_1} \cap \mathcal{Y}_0'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \mathcal{A}_{x_1} \cap \mathcal{O}\} && \text{(as } \mathcal{Y}_0 = \emptyset) \\ &= \min_z \{z \in \mathcal{A}_{x_1}\} = \pi(x_1) && \text{(from Eqn. (20a))} \end{aligned} \tag{21}$$

Hence, $y_1 \neq \emptyset$ and $\mathcal{Y}_1 = \{\pi(x_1)\}$ is the set of matched outputs after routing on the first input.

Similarly for $x_2 \in \mathcal{I}_S$ we get

$$\begin{aligned} y_2 &= \min_z \{z \in \mathcal{A}_{x_2} \cap (\mathcal{Y}_1)'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \{\pi(x_2)\} \cap \{\pi(x_1)\}'\} && \text{(from Eqn. (20a))} \\ &= \min_z \{z \in \{\pi(x_2)\}\} = \pi(x_2) && \text{(as } \pi(x_1) \neq \pi(x_2)) \end{aligned}$$

Continuing this way we get $\mathcal{Y}_k = \{\pi(x_1), \pi(x_2), \ldots, \pi(x_k)\}$ (22)

$\mathcal{X}$ is an ordered set of distinct inputs, which means that all the elements of $\mathcal{Y}_k$ are distinct and hence form a matching for the k slim inputs. If $k = r$ then the proof is complete, else for $x_{k+1} \in \mathcal{I}_F$ we get

$$\begin{aligned} y_{k+1} &= \min_z\{z \in \mathcal{A}_{x_{k+1}} \cap (\mathcal{Y}_k)'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \mathcal{O} \cap \{\pi(x_1), \ldots, \pi(x_k)\}'\} && \text{(from Eqns. (20b) and (22))} \\ &= \min_z\{z \in \{\pi(x_1), \ldots, \pi(x_k)\}'\} && (23) \end{aligned}$$

Note $|\{\pi(x_1), \ldots, \pi(x_k)\}'| = m - k$. Let $\mathcal{B} = \{b_1, \ldots, b_{m-k}\}$ where $b_i \in (\mathcal{Y}_k)',\ i = 1, \ldots, m - k,$ such that $b_1 < b_2 < \cdots < b_{m-k}$. Thus, $\mathcal{B}$ is an ordered version of $(\mathcal{Y}_k)' = \{\pi(x_1), \ldots, \pi(x_k)\}'$ with elements arranged in increasing order of magnitude, $|\mathcal{B}| = m - k$. From Eqn. (23), $y_{k+1} = \min_z\{z \in \mathcal{B}\} = b_1$.

Also, $k \leq r \leq m$ and if $k = r = m$, i.e., all the slim inputs are concentrated then $y_{k+1} = \emptyset$ and $\mathcal{Y}_k$ is the matching corresponding to the concentration. If $k < r$ then clearly $y_{k+1} = b_1 \neq \emptyset$ as then $\mathcal{B} \neq \emptyset$. Thus $\mathcal{Y}_{k+1} = \mathcal{Y}_k \cup \{b_1\} = \{\pi(x_1), \ldots, \pi(x_k), b_1\}$.

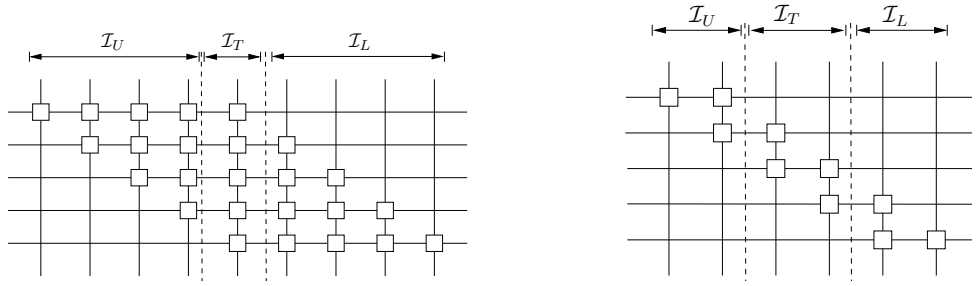
For input $y_{k+2} \in \mathcal{I}_F$ we get
$$\begin{aligned} y_{k+2} &= \min_z\{z \in \mathcal{A}_{x_{k+2}} \cap (\mathcal{Y}_{k+1})'\} \\ &= \min_z\{z \in \mathcal{O} \cap (\mathcal{Y}_k \cup \{b_1\})'\} \\ &= \min_z\{z \in (\mathcal{Y}_k)' \cap \{b_1\}'\} \\ &= \min_z\{z \in \mathcal{B} \cap \{b_1\}'\} = b_2 \end{aligned}$$

Continuing in the same fashion we get $\mathcal{Y}^r_{k+1} = \{b_1, \ldots, b_{r-k}\}$. Now, $|\mathcal{B}| = m - k$, i.e., $k = m - |\mathcal{B}|$. Thus, $y_{k+i} = b_i = b_{(k+i)-k} = b_{k+i-(m-|\mathcal{B}|)}$, $i = 1, \ldots, r - k$. Hence, $y_i = b_{i-(m-a)},\ i = k + 1, \ldots, r$, $a = |\mathcal{B}|$. Therefore, by Theorem 2 the fat-slim QSC(n, m) is self routing. □

5.3 Self-Routing Banded QSC(n, m)

We show that for banded sparse crossbar concentrators our self-routing scheme can find an r-output matching for any r input subset ($r \leq m$).

Definition 5 (Banded Crossbar)**.** An (n, m)-sparse crossbar is called banded if the set of inputs, $\mathcal{I}$, can be partitioned into three sets $\mathcal{I}_U, \mathcal{I}_T$ and $\mathcal{I}_L$ as shown in Figure 10 with the corresponding neighbor sets for the inputs as follows:



(a) Partitions of banded crossbar, $n = 9, m = 5$, $(n \geq 2m - 1)$.

(b) Partitions of banded crossbar, $n = 6, m = 5$, $(m \leq n < 2m - 1)$.

Fig. 10: Partitions of banded sparse crossbar concentrator.

If $n \geq 2m - 1$ then:

$$i \in \mathcal{I}_U \quad \Leftrightarrow \quad 1 \leq i \leq m - 1; \ \mathcal{A}_i = \{1, 2, \ldots, i\} \tag{24a}$$

$$i \in \mathcal{I}_T \quad \Leftrightarrow \quad m \leq i \leq n - m + 1; \ \mathcal{A}_i = \{1, 2, \ldots, m\} = \mathcal{O} \tag{24b}$$

$$i \in \mathcal{I}_L \quad \Leftrightarrow \quad n - m + 2 \leq i \leq n; \ \mathcal{A}_i = \{i - n + m, \ldots, m\} \tag{24c}$$

If $m \leq n < 2m - 1$ then:

$$i \in \mathcal{I}_U \quad \Leftrightarrow \quad 1 \leq i \leq n - m + 1; \ \mathcal{A}_i = \{1, 2, \ldots i\} \tag{25a}$$

$$i \in \mathcal{I}_T \quad \Leftrightarrow \quad n - m + 2 \leq i \leq m - 1; \ \mathcal{A}_i = \{i - n + m, \ldots, i\} \tag{25b}$$

$$i \in \mathcal{I}_L \quad \Leftrightarrow \quad m \leq i \leq n; \ \mathcal{A}_i = \{i - n + m, \ldots, m\} \tag{25c}$$

Note that for $n = 2m - 2$, $n - m + 2 = m > m - 1$. Hence in this case, from Eqn. (25b), $\mathcal{I}_T$ does not exist, but this does not affect the proof below. Also, Eqns. (24a)-(25c) can be written more succinctly as:

$$\mathcal{A}_i = \{\max(1, \ i - n + m), \ldots, \min(i, \ m)\}, \quad i = 1, \ldots, n \tag{26}$$

Every banded (n, m)-sparse crossbar is an optimal (n, m)-concentrator with $m(n - m + 1)$ cross-points [18].

Theorem 4. *For the banded QSC(n,m) let $\mathcal{X} = \{x_1, x_2, \ldots, x_r\}$ be any ordered r-input subset where $x_1 < x_2 < \cdots < x_r$, $\forall\, r, 1 \leq r \leq m$. Then, there exists an output matching, $\mathcal{Y} = \{y_1, y_2, \ldots, y_r\}$ for $\mathcal{X}$ obtained as result of self-routing the banded QSC(n, m) and it is given by*

$$y_i = \max(i, \ x_i - n + m) \quad i = 1, 2, \ldots, r.$$

Proof: In $\mathcal{X}$ let k_1 inputs belong to $\mathcal{I}_U$, k_2 inputs belong to $\mathcal{I}_T$ and the rest $r-(k_1+k_2)$ inputs belong to $\mathcal{I}_L$, i.e., $\{x_1,\ldots,x_{k_1}\} \subseteq \mathcal{I}_U$, $\{x_{k_1+1},\ldots,x_{k_1+k_2}\} \subseteq \mathcal{I}_T$ and $\{x_{k_1+k_2+1},\ldots,x_r\} \subseteq \mathcal{I}_U$.

Case I: $n \geq 2m-1$

For input $x_1 \in \mathcal{I}_U$:

$$\begin{aligned} y_1 &= \min_z\{z \in \mathcal{A}_{x_1} \cap \mathcal{Y}_0'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \mathcal{A}_{x_1} \cap \mathcal{O}\} && \text{(as } \mathcal{Y}_0 = \emptyset) \\ &= \min_z\{z \in \{1,2,\ldots,x_1\}\} = 1 && \text{(from Eqn. (24a))} \end{aligned}$$

Similarly for $x_2 \in \mathcal{I}_U$

$$\begin{aligned} y_2 &= \min_z\{z \in \mathcal{A}_{x_2} \cap (\mathcal{Y}_1)'\} \\ &= \min_z\{z \in \{1,\ldots,x_2\} \cap \{1\}'\} \\ &= \min_z\{z \in \{2,\ldots,x_2\}\} = 2 \end{aligned}$$

Proceeding this way for all inputs in $\mathcal{I}_U$ we get $\mathcal{Y}_{k_1} = \{1,2,\ldots,k_1\}$.

For input $x_{k_1+1} \in \mathcal{I}_T$:

$$\begin{aligned} y_{k_1+1} &= \min_z\{z \in \mathcal{A}_{x_{k_1+1}} \cap (\mathcal{Y}_{k_1})'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \{1,\ldots,m\} \cap \{1,\ldots,k_1\}'\} && \text{(from Eqn. (24b))} \\ &= \min_z\{z \in \{k_1+1,\ldots,m\}\} = k_1+1 \end{aligned}$$

We can get similar results for all other inputs in $\mathcal{I}_T$. Thus, $\mathcal{Y}_{k_1+k_2} = \{1,\ldots,k_1+k_2\}$.

Also when $1 \leq i \leq k_1+k_2$, $x_i \in \mathcal{I}_U \cup \mathcal{I}_T$

$$\begin{aligned} \Rightarrow \quad & x_i \leq n-m+1 && \text{(from Eqns. (24a) and (24b))} \\ \Rightarrow \quad & x_i - n + m \leq 1 \\ \Rightarrow \quad & x_i - n + m \leq i = y_i \end{aligned}$$

Hence, $y_i = i = \max(i,\ x_i - n + m)$, for all $1 \leq i \leq k_1+k_2$.

For input $x_{k_1+k_2+1} \in \mathcal{I}_L$:

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z\{z \in \mathcal{A}_{x_{k_1+k_2+1}} \cap (\mathcal{Y}_{k_1+k_2})'\} \\ &= \min_z\{z \in \{x_{k_1+k_2+1}-n+m,\ldots,m\} \cap \{k_1+k_2+1,\ldots,m\}\} && (27) \\ &= \min_z\{z \in \{\max(x_{k_1+k_2+1}-n+m,\ k_1+k_2+1),\ldots,m\}\} && (28) \\ &= \max(k_1+k_2+1,\ x_{k_1+k_2+1}-n+m) && (29) \end{aligned}$$

where Eqn. (27) follows from Eqn. (24c) and Eqn. (28) follows from the fact that we are taking the intersection of 2 sets both of which cover continuous intervals of outputs up to output m. We now use induction to prove the rest of the theorem.

Induction assumption:

$$y_i = \max(i,\ x_i - n + m), \quad \text{for all } i = k_1 + k_2 + 1, \ldots, j-1 \text{ where } j \leq r. \tag{30}$$

Need to prove: $y_j = \max(j,\ x_j - n + m)$

Proof for Induction: We have already proved the base case for $i = k_1 + k_2 + 1$. We will first show that, for y_i's chosen according to Eqn. (30), $y_{i-1} < y_i$.

Note $y_{i-1} = \max(i-1, x_{i-1} - n + m)$ and $y_i = \max(i, x_i - n + m)$, thus we get the following cases:

1) $y_{i-1} = i - 1$: We get the followings series of inequalities:

$$\begin{aligned} \max(i, x_i - n + m) &\geq i \\ \Rightarrow y_i &\geq i \quad (\text{as } y_i = \max(i, x_i - n + m)) \\ &> i - 1 = y_{i-1} \end{aligned}$$

2) $y_{i-1} = x_{i-1} - n + m$: We get the followings series of inequalities:

$$\begin{aligned} \max(x_i - n + m, i) &\geq x_i - n + m \\ \Rightarrow y_i &\geq x_i - n + m \quad (\text{as } y_i = \max(i, x_i - n + m)) \\ &> x_{i-1} - n + m = y_{i-1} \quad (\text{as } x_i > x_{i-1}) \end{aligned}$$

Thus $y_i > y_{i-1},\ i = k_1 + k_2 + 2, \ldots, j-1$.

We know that $\mathcal{Y}_{k_1+k_2} = \{1, \ldots, k_1 + k_2\}$ is monotonically increasing, $\mathcal{Y}_{k_1+k_2+1}^{j-1}$ is monotonically increasing and $y_{k_1+k_2+1} = \max(k_1 + k_2 + 1, x_{k_1+k_2+1} - n + m)) > k_1 + k_2 = y_{k_1+k_2}$.

Thus $\mathcal{Y}_{j-1} = \mathcal{Y}_{k_1+k_2} \cup \mathcal{Y}_{k_1+k_2+1}^{j-1}$ is monotonically increasing, i.e., $y_1 < y_2 < \cdots < y_{j-1}$.

By Lemma 2, $y_j = \min_z \{z \in \mathcal{A}_{x_j} \cap (\mathcal{Y}_{j-1})'\}$ (31)

By induction assumption $y_{j-1} = \max(j-1, x_{j-1} - n + m)$.

Case 1: $y_{j-1} = x_{j-1} - n + m$

By monotonicity of $\mathcal{Y}_{j-1}$ we get $\max_z(z \in \mathcal{Y}_{j-1}) = y_{j-1} = x_{j-1} - n + m$. Thus

$$(\mathcal{Y}_{j-1})' = \mathcal{Z} \ \cup \ \{x_{j-1} - n + m + 1, \ldots, m\}$$

$$\text{where } \mathcal{Z} \subseteq \{k_1 + k_2 + 1, \ldots, x_{j-1} - n + m - 1\}$$

Therefore,

$$\begin{aligned}
\mathcal{A}_{x_j} \cap (\mathcal{Y}_{j-1})' &= \{x_j - n + m, \ldots, m\} \cap [\mathcal{Z} \cup \{x_{j-1} - n + m + 1, \ldots, m\}] \\
&= \emptyset \cup \{x_j - n + m, \ldots, m\} \cap \{ x_{j-1} - n + m + 1, \ldots, m\} \\
&= \{(\max(x_j - n + m,\ x_{j-1} - n + m + 1)), \ldots, m\} \\
&= \{x_j - n + m, \ldots, m\} \quad (\text{as } x_j \geq x_{j-1} + 1) \qquad (32)
\end{aligned}$$

Substituting in Eqn. (31) we get

$$y_j = \min_z \{z \in \{x_j - n + m, \ldots, m\}\} = x_j - n + m \qquad (33)$$

Now $y_{j-1} = \max(x_{j-1} - n + m,\ j - 1) = x_{j-1} - n + m$. Thus

$$\begin{aligned}
x_{j-1} - n + m &\geq j - 1 \\
\Rightarrow x_{j-1} - n + m + 1 &\geq j \\
\Rightarrow x_j - n + m &\geq j \quad (\text{as } x_j \geq x_{j-1} + 1) \qquad (34)
\end{aligned}$$

From Eqns. (33) and (34)

$$y_j = \max(x_j - n + m,\ j) \qquad (35)$$

Case 2: $y_{j-1} = j - 1$

$\mathcal{Y}_{j-1}$ is monotonically increasing, i.e., $y_1 < y_2 < \cdots < y_{j-1}$ and $y_{j-1} = j - 1$

$$\begin{aligned}
\Rightarrow \mathcal{Y}_{j-1} &= \{1, 2, \ldots, j - 1\} \\
\Rightarrow (\mathcal{Y}_{j-1})' &= \{j, \ldots, m\}
\end{aligned}$$

Substituting in Eqn. (31) we get

$$\begin{aligned}
y_j &= \min_z \{\{x_j - n + m, \ldots, m\} \cap \{ j, \ldots, m\}\} \\
&= \min_z \{\max(x_j - n + m,\ j), \ldots, m\} \\
&= \max(x_j - n + m,\ j) \qquad (36)
\end{aligned}$$

From Eqn. (35) and Eqn. (36) $y_j = \max(x_j - n + m,\ j)$ and proof for the induction is complete.

Case II: $m \leq n < 2m - 1$

k_1 inputs in S belong to $\mathcal{I}_U$, i.e., $\{x_1, \ldots, x_{k_1}\} \subseteq \mathcal{I}_U$.

For $y_1, \ldots, y_{k_1}$ the proof is exactly the same as for the case $n \geq 2m - 1$ and we get $\mathcal{Y}_{k_1} = \{1, 2, \ldots, k_1\}$ and hence, $y_i = i = \max(i,\ x_i - n + m)$, $i = 1, \ldots, k_1$.

The next k_2 inputs in $\mathcal{X}$ belong to $\mathcal{I}_T$, i.e., $\{x_{k_1+1}, \ldots, x_{k_1+k_2}\} \subseteq \mathcal{I}_T$.

For y_{k_1+1} we get

$$\begin{aligned} y_{k_1+1} &= \min_z \{z \in \mathcal{A}_{x_{k_1+1}} \cap (\mathcal{Y}_{k_1})'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \{x_{k_1+1} - n + m, \ldots, x_{k_1+1}\} \cap \{1, \ldots, k_1\}'\} && \text{(from Eqn. (25b))} \\ &= \min_z \{z \in \{x_{k_1+1} - n + m, \ldots, x_{k_1+1}\} \cap \{k_1 + 1, \ldots, m\}\} \end{aligned}$$

Both the sets in the intersection contain a continuous series of outputs. Now obviously, $x_{k_1+1} \leq m$. Also, $k_1 + 1 \leq x_{k_1+1}$ as

$$\begin{aligned} k_1 &\leq n - m + 1 && \text{(from Eqn. (25a))} \\ \Rightarrow k_1 + 1 &\leq n - m + 2 \\ &\leq x_{k_1+1} && \text{(from Eqn. (25b))} \end{aligned}$$

Thus,
$$\begin{aligned} y_{k_1+1} &= \min_z \{z \in \{\max(k_1 + 1, x_{k_1+1} - n + m), \ldots, x_{k_1+1}\}\} \\ &= \max(k_1 + 1, x_{k_1+1} - n + m) \end{aligned}$$

Treating this as the base case, we can use an induction argument similar to that employed for inputs in $\mathcal{I}_L$ for the case $n \geq 2m - 1$ to show that

$$\begin{aligned} y_i &= \max(i,\ x_i - n + m), \quad k_1 + 1 \leq i \leq k_1 + k_2 \\ \Rightarrow y_i &= \max(i,\ x_i - n + m), \quad x_i \in \mathcal{I}_T \end{aligned} \tag{37}$$

For input $x_{k_1+k_2+1}$ we get

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z \{z \in \mathcal{A}_{x_{k_1+k_2+1}} \cap (\mathcal{Y}_{k_1+k_2})'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap (\mathcal{Y}_{k_1+k_2})'\} \end{aligned} \tag{38}$$

As $y_{k_1+k_2} = \max(k_1 + k_2,\ x_{k_1+k_2} - n + m)$, we get the following two cases:

Case 1: $y_{k_1+k_2} = k_1 + k_2$. Since $y_1 < y_2 < \cdots < y_{k_1+k_2}$, and $y_{k_1+k_2} = k_1 + k_2$

$$\begin{aligned} \mathcal{Y}_{k_1+k_2} &= \{1, 2, \ldots, k_1 + k_2\} \\ \Rightarrow (\mathcal{Y}_{k_1+k_2})' &= \{k_1 + k_2 + 1, \ldots, m\} \end{aligned} \tag{39}$$

Substituting Eqn. (39) in Eqn. (38) we get

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z\{z \in \{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap \{\, k_1 + k_2 + 1, \ldots, m\}\} \\ &= \min_z\{z \in \{\max(x_{k_1+k_2+1} - n + m,\ k_1 + k_2 + 1), \ldots, m\}\} \\ &= \max(x_{k_1+k_2+1} - n + m,\ k_1 + k_2 + 1) \end{aligned} \tag{40}$$

Case 2: $y_{k_1+k_2} = x_{k_1+k_2} - n + m$. Since $y_1 < \cdots < y_{k_1+k_2}$, $\max_z(z \in \mathcal{Y}_{k_1+k_2}) = y_{k_1+k_2} = x_{k_1+k_2} - n + m$.

$\Rightarrow (\mathcal{Y}_{k_1+k_2})' = \mathcal{Z} \cup \{x_{k_1+k_2} - n + m + 1, \ldots, m\},\ \mathcal{Z} \subseteq \{k_1 + k_2 + 1, \ldots, x_{k_1+k_2} - n + m - 1\}$.

Substituting this in Eqn. (40) we get

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z\{z \in \{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap (\mathcal{Z} \cup \{x_{k_1+k_2} - n + m + 1, \ldots, m\})\} \\ &= \min_z\{z \in \emptyset \cup (\{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap \{\, x_{k_1+k_2} - n + m + 1, \ldots, m\})\} \\ &= \min_z\{z \in \{\max(x_{k_1+k_2} - n + m + 1,\ x_{k_1+k_2+1} - n + m), \ldots, m\}\} \\ &= \max(x_{k_1+k_2} - n + m + 1,\ x_{k_1+k_2+1} - n + m) \\ &= x_{k_1+k_2+1} - n + m \qquad (\text{as } x_{k_1+k_2+1} \geq x_{k_1+k_2} + 1) \end{aligned} \tag{41}$$

Also,

$$\begin{aligned} x_{k_1+k_2+1} - n + m &\geq x_{k_1+k_2} + 1 - n + m \\ &= y_{k_1+k_2} + 1 \\ &\geq k_1 + k_2 + 1 \end{aligned} \tag{42}$$

where Eqn. (42) follows from the fact that $y_{k_1+k_2} = \max(x_{k_1+k_2} - n + m,\ k_1 + k_2)$. From Eqns. (41) and (42), $y_{k_1+k_2+1} = \max(k_1 + k_2 + 1,\ x_{k_1+k_2+1} - n + m)$.

We can now use an induction argument for rest of the inputs in $\mathcal{I}_L$ similar to the case for $n \geq 2m - 1$ to show that

$$\begin{aligned} y_i &= \max(i,\ x_i - n + m), \quad k_1 + k_2 + 1 \leq i \leq r \\ \Rightarrow y_i &= \max(i,\ x_i - n + m), \quad x_i \in \mathcal{I}_L. \end{aligned}$$

Thus, by Theorem 2 the banded QSC(n, m) is self-routing. □

6 An Example

We give an example to illustrate self-routing on a fat-slim QSC$(5, 3)$. This concentrator is shown in Figure 11. The quantum packets present at inputs $1, 3$ and 4 are $|Q_1\rangle = \frac{1}{\sqrt{2}}(|1, d_{11}\rangle + |1, d_{12}\rangle)$,

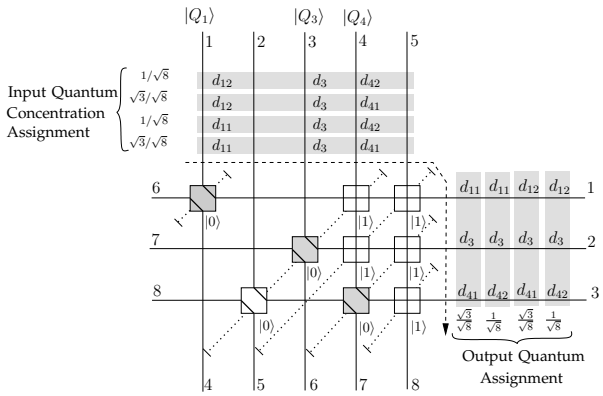


Fig. 11: Self-Routing on fat-slim QSC(5,3).

$|Q_3\rangle = |1, d_3\rangle$ and $|Q_4\rangle = \frac{\sqrt{3}}{2}|1, d_{41}\rangle + \frac{1}{2}|1, d_{42}\rangle$ respectively. Inputs 2 and 5 do not have any packets. Inputs $6, 7$ and 8 correspond to the three dummy inputs on the left hand side from top to bottom. Thus, in this case, the input quantum concentration assignment is given by

$$
\begin{aligned}
& |Q_1\rangle \otimes |0, d_2\rangle \otimes |Q_3\rangle \otimes |Q_4\rangle \otimes |0, d_5\rangle \bigotimes_{i=6}^{8} |0, d_i\rangle \\
= & \left(\frac{1}{\sqrt{2}}|1, d_{11}\rangle + \frac{1}{\sqrt{2}}|1, d_{12}\rangle\right) \otimes |0, d_2\rangle \otimes |1, d_3\rangle \otimes \left(\frac{\sqrt{3}}{2}|1, d_{41}\rangle + \frac{1}{2}|1, d_{42}\rangle\right) \\
& \qquad \otimes |0, d_5\rangle \otimes |0, d_6\rangle \otimes |0, d_7\rangle \otimes |0, d_8\rangle \\
= & \frac{\sqrt{3}}{2\sqrt{2}}|(1, d_{11}), (0, d_2), (1, d_3), (1, d_{41}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \\
& + \frac{1}{2\sqrt{2}}|(1, d_{11}), (0, d_2), (1, d_3), (1, d_{42}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \\
& + \frac{\sqrt{3}}{2\sqrt{2}}|(1, d_{12}), (0, d_2), (1, d_3), (1, d_{41}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \\
& + \frac{1}{2\sqrt{2}}|(1, d_{12}), (0, d_2), (1, d_3), (1, d_{42}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \qquad (43)
\end{aligned}
$$

Thus, the input is a superposition of four concentration patterns with co-efficients $\sqrt{3}/\sqrt{8}$, $1/\sqrt{8}, \sqrt{3}/\sqrt{8}$ and $1/\sqrt{8}$ respectively, shown by grey horizontal bars. Since all four patterns are capacity achieving, the quantum assignment is also capacity achieving. The state of the crosspoints is also shown. The shaded crosspoints route the valid packets on inputs $1, 3$ and 4. Measurement at the output will result in one out of the four patterns shown at the output being observed with probabilities $3/8, 1/8, 3/8$ and $1/8$ respectively. Therefore, data packets d_{11} and d_{12} are observed on output 1 with probability $1/2$. Data packet d_3 is observed on output 2 with probability 1 and data packets d_{41} and d_{42} are observed with probability $3/4$ and $1/4$

respectively on output 3. This output state can be explicitly written as:

$$\begin{aligned}
&\frac{\sqrt{3}}{2\sqrt{2}}\,|(1,d_{11}),(1,d_3),(1,d_{41}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \\
&\quad+\frac{1}{2\sqrt{2}}\,|(1,d_{11}),(1,d_3),(1,d_{42}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \\
&\quad+\frac{\sqrt{3}}{2\sqrt{2}}\,|(1,d_{12}),(1,d_3),(1,d_{41}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \\
&\quad+\frac{1}{2\sqrt{2}}\,|(1,d_{12}),(1,d_3),(1,d_{42}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \qquad (44)
\end{aligned}$$

The packets in the concentration patterns are written in increasing order of outputs, with outputs 1–3 on the right and outputs 4–8 located on the bottom. The dashed arrow shows the order in which the crosspoints are traversed during routing, and the crosspoints in one stage are indicated by the dotted diagonals. The initial state of the auxiliary qubits (control qubits) is $|000000000\rangle$ which is a string of nine zeros, each corresponding to one crosspoint in the crossbar. Recall that control qubits are set to 1 for the cross state and to 0 for the through state. The output state of the auxiliary qubits is indicated beside the crosspoints. Thus, the output state of the auxiliary qubits is $|000110111\rangle$ where the bits are written in order from top to bottom and left to right, e.g., the third crosspoint for input 4 is set to a through state and this is the sixth crosspoint in traversal order, so the sixth bit in the output state is 0.

6.1 Output for Capacity Exceeding Input Patterns

So far we have shown that in a fat-slim or banded QSC(n,m), self-routing can be used to concentrate any capacity achieving input assignment pattern. We now present the case when the input pattern exceeds the capacity of the crossbar.

For a self-routing QSC(n,m), consider a capacity exceeding input concentration pattern with r, $(r>m)$ valid packets. The ordered set of inputs with packets to concentrate is $\mathcal{X}=\mathcal{X}_m\cup\mathcal{X}_{m+1}^r$ where $\mathcal{X}_m=\{x_1,\ldots,x_m\}$ and $\mathcal{X}_{m+1}^r=\{x_{m+1},\ldots,x_r\}$. Since inputs are routed in increasing order, all inputs in $\mathcal{X}_m$ are concentrated to the m outputs, i.e., $\mathcal{Y}=\{y_1,\ldots,y_m\}=\mathcal{O}$. For input x_{m+1}: $y_{m+1}=\min_z\{z\in\mathcal{A}_{x_{m+1}}\cap(\mathcal{Y})'\}=\min_z\{z\in\mathcal{A}_{x_{m+1}}\cap\emptyset\}=\emptyset$. Similarly for the other inputs in $\mathcal{X}_{m+1}^r$, the matching output is $\emptyset$, i.e., $\{y_{m+1},\ldots,y_r\}=\emptyset$. If $y_i=\emptyset$, then all crosspoints in the column for the corresponding input x_i are set to cross state and the packet comes out on the bottom, which is at output x_i+m. Hence, the m lowest numbered inputs are concentrated and the rest are connected to corresponding output at the bottom.

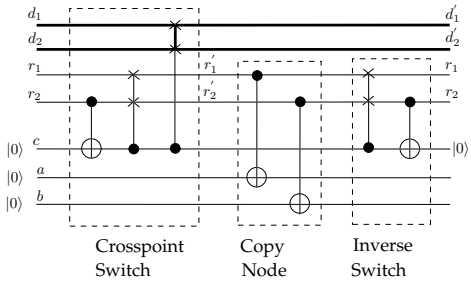


Fig. 12: Circuit for restoring the control quantum bit c.

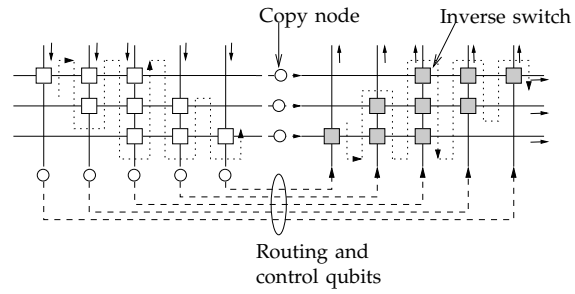


Fig. 13: A Banded QSC(5,3) with additions for restoring the control quantum bits.

7 Restoring Auxiliary Control Quantum Bits

Quantum information can be encoded in many different ways, such as the spin component of basic particles like electrons or protons, or in the polarization of photons. But, such particles can interact with the environment which leads to a corruption of their quantum state, a process known as *decoherence*. Decoherence can be viewed as a measurement of a superposed quantum state which collapses it to one of its basis states. This leads to a loss of information, but for a quantum circuit, this information loss can be overcome if the ancillary qubits used as control qubits are restored back to their original states, so that a corruption of their state does not affect the observed quantum data. We now give a method to restore the state of the auxiliary bits back to their original state, i.e., $|0\rangle$. For a single quantum crosspoint we can restore the control quantum bit back to the state $|0\rangle$ as shown in Figure 12. The mapping performed is:

$$|(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |00\rangle_{ab} \xrightarrow[Through]{Crosspoint} |(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |00\rangle_{ab} \tag{45}$$

$$\xrightarrow{Copy} |(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |r_1 0\rangle_{ab} \tag{46}$$

$$\xrightarrow{Inverse} |(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |r_1 0\rangle_{ab} \tag{47}$$

$$|(r_1, d_1), (1, d_2)\rangle\, |0\rangle_c\, |00\rangle_{ab} \xrightarrow[Cross]{Crosspoint} |(1, d_2), (r_1, d_1)\rangle\, |1\rangle_c\, |00\rangle_{ab} \tag{48}$$

$$\xrightarrow{Copy} |(1, d_2), (r_1, d_1)\rangle\, |1\rangle_c\, |1\, r_1\rangle_{ab} \tag{49}$$

$$\xrightarrow{Inverse} |(r_1, d_2), (1, d_1)\rangle\, |0\rangle_c\, |1\, r_1\rangle_{ab} \tag{50}$$

At the output of the quantum crosspoint the two CNOT gates in the copy circuit copy the values of bits r_1 and r_2 onto a and b respectively. This can be seen in Eqn. (46) and Eqn. (49). The inverse switch then does a controlled swap of the two routing bits r_1 and r_2 before restoring c back to its original state as can be seen in Eqn. (47) and Eqn. (50). Note that when the bit c

is 0 at the output of the quantum crosspoint switch then the restoring portion does not modify anything (see Eqns. (45)-(47)) as the corresponding auxiliary qubit is already in the state $|0\rangle$.

On measurement at the output we determine valid (not valid) packets by observing their associated routing bit as 1 (0). But note that on final measurement in Figure 12 the routing bits may not correspond to the data part of their packets, this is seen in Eqn. (50) where r_1, d_2 and $1, d_1$ are together instead of being $1, d_2$ and r_1, d_1. But the copying operation ensures that we have a copy of the correct routing bits and can use these to distinguish the valid packets, for example, in Eqn. (50) the correct values of the routing bits at the output are 1 for the upper packet and r_1 for the lower packet and these are present in the correct order, $1, r_1$, on qubits a and b. Thus we can now consider a as the routing qubit for the packet at the upper output and b as the routing qubit for the packet at the lower output.

This circuit restores the control qubit for a single crosspoint. For the entire self-routing QSC we need a mirror image of the sparse crossbar concentrator concatenated with the QSC after the copying of the routing bits is done at the output to restore the control qubits. This is shown in Figure 13. Only the routing qubits are involved in restoring the state of the control qubits, hence only these qubits are forwarded to the next stage after the QSC and are shown by dashed lines. The dotted lines show the order of traversal of crosspoints and inverse switches.

8 Concluding Remarks

We have introduced quantum concentrators and presented two designs of such concentrators using self-routing sparse crossbars. The complexity of these quantum concentrators can be computed as follows:

We need, per crosspoint, one multi-qubit switch gate for swapping the $n_d + 1$ bit packets and one CNOT gate for setting the auxiliary control bit. A switch gate for swapping one quantum bit packets can be implemented using two CNOT gates and one CCNOT gate. Hence, we need $2(n_d + 1)$ CNOT and $n_d + 1$ CCNOT gates for the multi-qubit switch gate for a total of $2n_d + 3$ CNOT and $n_d + 1$ CCNOT gates per crosspoint. Therefore, each quantum concentrator design we presented uses $m(n - m + 1)(2n_d + 3)$ CNOT gates, $m(n - m + 1)(n_d + 1)$ CCNOT gates and $m(n - m + 1)$ auxiliary quantum bits.

For the restoring stage, there are $m(n - m + 1)$ inverse switches and $n + m$ copy nodes. Each inverse switch has one switch gate for single qubits and one CNOT gate, which sums up to three CNOT and one CCNOT gate. Each copy node has two CNOT gates and two extra qubits. Thus the total cost for restoring the control qubits is $3m(n - m + 1) + 2(n + m)$ CNOT

gates, $m(n-m+1)$ CCNOT gates and $2(m+n)$ extra qubits. Therefore, the overall cost for a QSC(n,m) is $m(n-m+1)(2n_d+6)+2(n+m)$ CNOT gates, $m(n-m+1)(n_d+2)$ CCNOT gates and $m(n-m+1)+2(n+m)$ auxiliary quantum bits.

The depth of a QSC(n,m) is given by the maximum possible number of crosspoints between an input and an output. It is easy to see that the longest input-output path is between input 1 and output m. For the fat-slim QSC(n,m), this path length is $(n-m+1)+m-1=n$ crosspoints and for the banded QSC(n,m) the path length is $(n-m+1)+(m-1)+(m-1)=n+m-1$ crosspoints. Hence the depth of fat-slim QSC(n,m) is n and the depth of banded QSC(n,m) is $n+m-1$. The time required to self-route is upper bounded by the depth of the concentrator, thus, self-routing on a fat-slim QSC(n,m) has $O(n)$ delay and self-routing on a banded QSC(n,m) has $O(n+m)$ delay.

Our results demonstrate that quantum principles can be applied to concentration problems in packet switching. In proving that fat-slim and banded crossbar concentrators are self-routable, we have assumed as input quantum assignments consisting of single patterns of classical packets. This proves that classical fat-slim and banded crossbar concentrators are self-routable as well. We also note that in our self-routing algorithm, when the capacity of m packets is exceeded, only the m lowest numbered inputs have their packets concentrated. This introduces the issue of fairness in routing. One way to ensure all inputs have an equal chance to be concentrated when capacity is exceeded, is to create an equal superposition of the input packets at the outputs of a crosspoint when both input packets have their routing bits equal to 1. The resulting output quantum concentration assignment would then contain superposed concentration patterns in which valid packets from inputs other than the lowest m are present.

Another direction to be explored further is the tradeoff between delay and fanout. We see that the delay varies with topology, as the more "balanced" banded crossbar having a larger delay of $O(n+m)$ than the $O(n)$ delay for the fat-slim crossbar. This seems to be a consequence of the sequential nature of the routing algorithm. The dependence of delay on routing strategy and topology is thus another direction for further research.

We have proved the self-routability of two families of sparse crossbar concentrators, namely the fat-slim and banded crossbars. Finding other topologies which allow self-routing remains an open question. The density of (n,m)-sparse crossbar concentrators among all $n\times m$ crossbars is known [25] as are equivalence relations between different classical sparse crossbar concentrator topologies [18]. A similar approach could be employed to characterize self-routable crossbar

concentrators and find structures which belong to this class. In our algorithm, inputs were routed in increasing order of their index. By changing the order in which this routing is done, we could find other structures which allow self-routing. A trivial example would be to route in decreasing order of inputs on a fat-slim crossbar with inputs 1 to $(n-m)$ comprising the fat section and inputs $(n-m+1)$ to n the slim section. In particular, an interesting direction for further investigation would be to determine if there exist self-routing regular sparse crossbar concentrators, i.e., those with fixed out-degree inputs and in-degree outputs.

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, June 1999.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of STOC'96*, May 1996, pp. 212–219.

[3] D. Vion *et al.*, "Manipulating quantum state of an electrical circuit," *Science*, vol. 296, 2002.

[4] B. Kane, "A silicon-based nuclear spin quantum computer," *Nature*, vol. 393, 1998.

[5] D. Copsey *et al.*, "Toward a scalable, silicon-based quantum computing architecture," *IEEE Journal Of Selected Topics in Quantum Electronics*, vol. 9, no. 6, pp. 1552–1569, Nov-Dec 2003.

[6] M. Oskin *et al.*, "Building quantum wires: the long and the short of it," in *Proceedings of ISCA'03*, June 2003, pp. 374–385.

[7] W. Wootters and W. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, 1982.

[8] N. Isailovic *et al.*, "Datapath and control for quantum wires," *ACM Transactions on Architecture and Code Optimization*, vol. 1, no. 1, pp. 34–61, 2004.

[9] T. S. Metodi *et al.*, "A quantum logic array microarchitecture: Scalable quantum data movement and computation," in *Proceedings of MICRO-38*, 2005, pp. 305–318.

[10] I. M. Tsai and S.-Y. Kuo, "Digital switching in the quantum domain," *IEEE Transactions on Nanotechnology*, vol. 1, no. 3, pp. 154–164, 2002.

[11] M. K. Shukla, R. Ratan, and A. Y. Oruç, "A quantum self-routing packet switch," in *Proceedings of CISS'04*, Princeton, NJ, 2004, pp. 484–489.

[12] ——, "The quantum baseline network," in *Proceedings of CISS'05*, Baltimore, March 2005.

[13] S. T. Cheng and C. Y. Wang, "Quantum switching and quantum merge sorting," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 2, pp. 316–325, 2006.

[14] R. Ratan and A. Y. Oruç, "Quantum switching networks with classical routing," in *Proceedings of CISS'07*, Baltimore, March 2007.

[15] L. A. Bassalygo, "Asymptotically optimal switching circuits," *Problems of Information Transmission*, vol. 17, no. 3, pp. 206–211, 1981.

[16] L. A. Bassalygo and M. S. Pinsker, "Complexity of an optimum nonblocking switching network without reconnections," *Problems of Information Transmission*, vol. 9, no. 1, 1974.

[17] A. Yavuz Oruç and H. M. Huang, "Crosspoint complexity of sparse crossbar concentrators," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1466–1471, 1996.

[18] W. Guo and A. Y. Oruç, "Regular sparse crossbar concentrators," *IEEE Transactions on Computers*, vol. 47, no. 3, pp. 363–368, 1998.

[19] W. Guo, "Design and optimization of switching fabrics for ATM networks and parallel computer systems," Ph.D. dissertation, University of Maryland, August 1996.

[20] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, September 2000.

[21] D. Deutsch, "Quantum computational networks," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 425, no. 1868, pp. 73–90, Sep. 8 1989.

[22] J. Preskill. (1998) Physics 229: Advanced mathematical methods of physics – quantum computation and information. [Online]. Available: http://www.theory.caltech.edu/people/preskill/ph229

[23] M. K. Shukla and A. Y. Oruç, "Multicasting in quantum networks," 2008, Working draft—To be submitted to IEEE Transactions on Computers.

[24] S. Nakamura and G. M. Masson, "Lower bounds on crosspoints in concentrators," *IEEE Transactions on Computers*, vol. C-31, no. 12, pp. 1173–1179, 1982.

[25] E. Gündüzhan and A. Y. Oruç, "Structure and density of sparse crossbar concentrators," in *DIMACS Series in Discrete Mathematics and Computer Science*. Advances in Switching Networks, 1998, vol. 42, pp. 169–180.

Self-Routing Quantum Sparse Crossbar Packet Concentrators

Rahul Ratan, A. Yavuz Oruç
Department of Electrical and Computer Engineering,
University of Maryland, College Park, MD 20742

Abstract

Quantum switching networks are derived from conventional switching networks by replacing the classical switches by quantum switches. We give the quantum circuit design and routing of an $n \times m$ network, called a quantum concentrator that can direct quantum bit packets in arbitrary quantum states from any of its k inputs to some of its k outputs, where $1 \leq k \leq m \leq n$. Our designs are based on sparse crossbars which are rectangular grids of 2×2 crosspoints. Sparse crossbar concentrator structures with theoretically minimum crosspoint complexities for any values of n and m are well-known but no self-routing algorithms have been reported for such concentrators. We transform two such families of optimal concentrators, called fat-slim and banded sparse crossbars into quantum networks and provide self-routing algorithms for these families of concentrators. In this process we extend the notion of packet concentration to a quantum network and design self-routing quantum crosspoints from quantum gates. We address issues critical to quantum operation like reversibility and localized self-routing and give a rigorous proof that quantum fat-slim and banded sparse crossbar concentrators are self-routable. The self-routing algorithms described in the paper can be used for both quantum and classical sparse crossbar concentrators by the linearity property of all quantum systems.

1 Introduction

The unique properties of quantum systems as manifested in the form of quantum parallelism and entanglement have been used in finding efficient solutions for classically intractable problems [1], [2], [3], [4]. Any realistic scenario for the future scaling up of quantum systems involves spatially distributed quantum devices which can interact with each other. Several recent schemes for large scale quantum computer architectures based on solid-state silicon have been proposed [5], [6] and are projected to provide the scalability required to achieve a useful computational substrate. The issue of quantum data transport has been recognized as a particularly critical requirement in upcoming silicon-based quantum computing technologies [7], [8]. Spatially distributed components introduce the need for *quantum wires* over which quantum data can be transmitted but building quantum wires and transferring quantum bits (qubits) is a non-trivial operation since, in general, quantum bits cannot be copied as a consequence of

the no-cloning theorem [9]. Proposals for building quantum wires in solid state technologies include the quantum swapping and teleportation based architectures in [8], [7], [10]. The high cost of such wires implies that the $O(n^2)$ wires needed to interconnect n quantum devices can be a major bottleneck in implementing quantum systems.

This cost can be greatly reduced by using efficient switching network designs. The basic premise behind this idea is to use arrangements of reconfigurable switches with input quantum bits on their own quantum wires and then route them to the required destination. These switches are represented using quantum circuits composed of quantum gates. In addition to reducing wire count, reconfigurable quantum switches can form integral parts of the quantum data distribution system in envisioned architectures for scalable quantum computing. For example, in the Quantum Logic Array (QLA) microarchitecture proposed in [11], the high-level quantum computer structure consists of logical quantum bits connected with a programmable communication network in which integrated switch islands are used to redirect quantum data from nearby logical quantum bits.

The first result relating quantum circuits to permutations of quantum bits appears to be given by Moore and Nilsson [12]. They presented a procedure by which a 6-stage or a 4-stage quantum circuit with the use of ancillary quantum bits can be formed to realize any permutation of n qubits. Although related to switching, Moore and Nilsson's procedure cannot be viewed as a quantum interconnection network as each choice of permutation is mapped to a different quantum circuit by their procedure. Tsai and Kuo [13] used this approach to map unicast and multicast assignments to quantum circuits consisting of C-NOT gates. The first quantum switch network using reconfigurable switches to route groups of quantum bits was given by Shukla et. al. in [14], [15]. This quantum network can permute quantum information packets between its n inputs and n outputs. It was shown that this network realizes $n^{n/2}$ permutations and can be used to resolve blocking when transmitting classical packets by creating a superposition of packets whenever they contend for the same wire in the network. Cheng and Wang [16] proposed a quantum merge sorting-based network using $O(n \log^3 n)$ gates, while Sue [17], [18] gave the design of another quantum nonblocking network with $O(n^2)$ gates. Both of these networks can realize all $n!$ permutations. Switching network configurations suitable for quantum networks have also been identified in [19] and Cheng et. al. [20] presented an application of quantum routing to wireless networks.

All these networks realize ordered connection maps, i.e., the particular output to which an

input is connected is specified beforehand. In this paper we focus on the quantum circuit design and routing of concentrator switches that realize unordered connections between a set of inputs and a set of outputs. Even though concentrator switches are not as versatile as those that can realize ordered connections, classical analogs of such switches play a fundamental role in the design of nonblocking switching networks [21], [22]. More recently, it has been shown that they can also be used to design quantum multicast switching networks [29].

Formally, an (n, m)-concentrator is a switching network with n inputs and m outputs, $1 \leq m \leq n$ in which any set of k inputs can be routed over non-intersecting paths to some k outputs, $1 \leq k \leq m$, but without the order specified. This is unlike other switches such as permutation networks that provide ordered connections between their inputs and outputs. Switching networks can exhibit blocking where multiple input-output connections can share the same path internally. The stronger requirement of ordered connections makes this problem more severe, resulting in higher routing and hardware complexity. For example, the theoretical lower bound for hardware complexity of permutation networks is $O(n \log n)$ [23] while that for concentrators is $O(n)$ [22].

A family of concentrators, called *sparse crossbar concentrators*, can be designed using a grid or matrix of crosspoints with m rows and n columns where a crosspoint is placed between column i and row j if there exists an edge between input i and output j. Explicit sparse crossbar concentrator structures with theoretically minimum crosspoint complexities ($O(m(n-m+1))$) for any arbitrary values of n and m are well-known [24], [25]. The main contribution of this paper is to transform two such families of optimal concentrators, called fat-slim and banded sparse crossbars into quantum networks and provide self-routing algorithms for these families of concentrators.

We first give an interpretation of concentration in a quantum network. In a classical concentrator, packets for concentration are marked at inputs and these packets can be mapped only one assignment at a time. In a quantum concentrator, packets consist of quantum bits and thus represent a superposition of assignment patterns of packets which can be concentrated all at once by such a network due to the principle of quantum parallelism. This aspect is what distinguishes quantum concentrators from their classical counterparts. For example, consider a concentrator in which three inputs, say X, Y and Z, have packets which have to be concentrated. Suppose input X has two packets represented as x_1 and x_2, input Y has one packet y_1 and input Z has two packets z_1 and z_2. Y generates a "pure" packet while X and Z generate quantum

packets by creating a superposition of both their respective packets, and all three input sources then push their packets into a quantum concentrator. The outcome is that all the four possible input packet patterns: (x_1, y_1, z_1), (x_1, y_1, z_2), (x_2, y_1, z_1), (x_2, y_1, z_2) are routed in parallel and the output is a superposition of these four packet patterns each of which corresponds to the output obtained by concentrating one of the input packet patterns.

In the process of designing a quantum concentrator, we address some issues particular to quantum systems. One such issue is the reversibility constraint of quantum information processing. All quantum operations are inherently reversible in nature. This notion of reversibility is exactly the same as that commonly understood for any input-output mapping, i.e., given the output state of a quantum system, the corresponding input state can be uniquely determined. A "rectangular" structure like an (n, m)-sparse crossbar concentrator where the number of inputs, n, is not equal to the number of outputs, m, is inherently non-reversible. We devise a way to make (n, m)-crossbar concentrators "square" by using additional lines on the input and output sides of such concentrators and ensuring that valid packets for concentration are routed only among the original n inputs and m outputs.

Additionally, in a concentrator, a subset of inputs, say $\mathcal{I}_s$ can, in general, be matched to multiple subsets of outputs and a subset of outputs, say $\mathcal{O}_s$ can be the matching for multiple input subsets. Even when $\mathcal{O}_s$ is the only matching for $\mathcal{I}_s$, there may be multiple settings for the internal crosspoints which realize this matching. As a simple example, consider an (n, m)-concentrator in which a k-input subset and a k-output subset are interconnected by a $k \times k$ full crossbar, $k \leq m$. Also assume that the k inputs in the k-input subset are not connected to any other outputs. Then obviously this k-input subset can be matched to only one output set of size k but all possible $k!$ one-to-one maps are possible.

Thus, to ensure reversibility a routing algorithm is needed to determine the crosspoint settings and fix the output matching subset for a given input subset. Even though efficient centralized routing algorithms for optimal crossbar concentrators with $O(\log n)$ delay for a tree-connected set of n processors and $O(n \log n)$ delay for a single processor are known [25], [26], they cannot be adopted for quantum concentrators. It is critical to have a self-routing algorithm for quantum concentration in which the state of a crosspoint is determined by using only the local information from the incoming packet headers. An important property of self-routing packets is that the control quantum bits used to configure the crosspoint settings can be restored back to their original states easily thus preventing loss of information due to decoherence.

The rest of this paper is organized as follows. In Section 2, we introduce the basic quantum circuit concepts that will be used in the design and routing of quantum concentrators. In Section 3, we give a brief overview of classical sparse-crossbar concentrators. In Section 4 we present a quantum packet concentrator model that will be used to describe our self-routing algorithms. In Section 5, we define the functionality of quantum sparse-crossbar concentrators, present the design of quantum sparse crossbar concentrators and describe a self-routing algorithm for such concentrators. In Section 6, we prove the correctness of this algorithm for the optimal banded and fat-slim quantum crossbar concentrators. In Section 7, we give an example to describe the concentration process on a quantum sparse crossbar concentrator and address the issue of routing more than m packets on an (n, m)-quantum crossbar concentrator. In Section 8, we describe how to restore the state of auxiliary qubits to prevent decoherence. Section 9 gives the cost analysis and the paper is concluded with remarks on remaining questions and future work in Section 10.

2 Quantum Circuits and Packets

In this section we give a brief description of basic concepts related to quantum information, quantum circuits, quantum gates, and quantum packets.

2.1 Qubits

The indivisible unit of classical information is the bit: an object that can take either one of two values: 0 or 1. The corresponding unit of quantum information is the quantum bit or *qubit*. Unlike a classical bit, a qubit can take values which are, in some sense, a combination of the values 0 and 1, i.e., it can be simultaneously be both 0 and 1. Formally, a qubit's state is represented as a unit vector in a two-dimensional complex Hilbert space and is expressed as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad |\alpha|^2 + |\beta|^2 = 1;\ \alpha, \beta \in \mathbb{C} \tag{1}$$

The vectors $|0\rangle$ and $|1\rangle$ constitute an orthonormal basis for this space. These two vectors are referred to as the computational basis vectors. We can perform a measurement that projects the state of this qubit onto the computational basis, i.e., the measurement projects $|\psi\rangle$ onto the basis $\{|0\rangle\,, |1\rangle\}$. The outcome of the measurement is not deterministic— the probability that we observe the result to be $|0\rangle$ is $|\alpha|^2$ and the probability that we observe the result to be $|1\rangle$ is $|\beta|^2$. α and β are referred to as the probability amplitudes of the states $|0\rangle$ and $|1\rangle$ respectively.

The state of an n-qubit system can be expressed as vector in a complex Hilbert space of dimension 2^n. This 2^n dimensional space is a tensor product of the n two-dimensional spaces

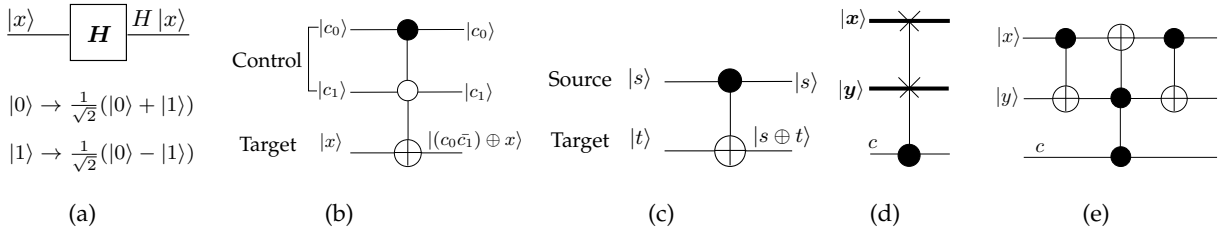


Fig. 1: Quantum gates: (a) Hadamard gate (b) Controlled-controlled not gate (c) Controlled-not (CNOT) gate (d) Controlled swap or switch gate (e) Switch gate made using CNOT gates.

representing individual qubits. The orthonormal basis for this space can be chosen as the states in which each qubit has a definite value, either $|0\rangle$ or $|1\rangle$. A general normalized vector representing an n-qubit state can be expanded in this basis as

$$\begin{aligned} |\psi\rangle &= \alpha_0 |00\cdots00\rangle + \alpha_1 |00\cdots01\rangle + \cdots + \alpha_{2^n-2} |11\cdots10\rangle + \alpha_{2^n-1} |11\cdots11\rangle \\ &= \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \end{aligned} \tag{2}$$

where we have associated with each string the number that it represents in binary notation, ranging in value from 0 to $2^n - 1$. Here the α_i's are complex numbers satisfying $\sum_i |\alpha_i|^2 = 1$. A measurement of all n qubits by projection of each onto the $\{|0\rangle, |1\rangle\}$ basis, yields the outcome $|i\rangle$ with probability $|\alpha_i|^2$ [4].

2.2 Quantum Gates

The state of qubits is transformed using quantum gates and circuits composed of such gates. The quantum gate formalism was first proposed by Deutsch [27]. A quantum gate is a linear, unitary transformation on the space of qubit state vectors. The unitary nature of these transformations implies that quantum gates are *reversible*, i.e., given the state of qubits at the output of a gate, the input state can be uniquely determined. The unitarity also implies that the gates preserve the norm of the input state which amounts to preserving probability. These requirements of reversibility and norm preservation are basic axioms of quantum theory.

An example of a single qubit gate is the Hadamard gate, H, (Figure 1(a)) which transforms the basis vectors $|0\rangle$ and $|1\rangle$ as

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \; |1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{3}$$

In the above mapping we say that the basis vectors $|0\rangle$ and $|1\rangle$ are put in an equal superposition by the Hadamard gate, as after the transformation, the probability of observing either of the

basis vectors at the output is equal to 1/2. Thus, the Hadamard gate can be considered a quantum randomizer which takes a 0 or 1 bit and transforms it so that the output is either 0 or 1 with probability 1/2 [3].

A gate is completely specified by the mapping it performs on the basis vectors as all the rest of the input states can be represented as vectors which are a linear combination of these basis vectors. In the case of the Hadamard gate this means that an input qubit in the general state $\alpha|0\rangle + \beta|1\rangle$ would be transformed to $\frac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\beta}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$.

An n-bit quantum circuit can simultaneously act on all the 2^n components of the input state and transform them according to some specified mapping at once. This is the source of massive *quantum parallelism*. To make this more clear, suppose we are interested in finding the properties of a function $f(i)$ which takes the n-bit binary string i as input. The table of values for $f(i)$ is of size 2^n and is clearly infeasible to calculate for large n. But, with a quantum computer, U_f acts according to

$$|i\rangle|0\rangle \xrightarrow{U_f} |i\rangle|f(i)\rangle \tag{4}$$

When we write any two qubit states side-by-side, it means we are taking a tensor product, thus $|i\rangle|0\rangle = |i\rangle \otimes |0\rangle$. We can put the input register consisting of the qubits corresponding to i in a superposed state similar to the one in Eqn. (2):

$$\underbrace{\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)}_{n \text{ qubits}} = \frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle \tag{5}$$

where we have taken the tensor product of the n qubits to get the complete state. By computing $f(i)$ only once, we can generate a state

$$\frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle|0\rangle \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle|f(i)\rangle \tag{6}$$

All the global properties of f are encoded in this state and can be extracted if an efficient method is devised. This is the kind of *massive parallelism* Shor used in his factorization algorithm [1]. This same parallelism enables us to probabilistically combine packets in quantum switching networks. The input to the quantum switching network can be a superposition of multiple packet assignments, all of which are routed in parallel to the outputs. The switching network operates in different switch configurations for different packet assignments. We give a more detailed explanation of this concept later in the paper for quantum concentrators.

Among all the gates which operate on multiple qubit inputs, the most widely-used gates are the controlled quantum gates. A controlled gate's function is determined by the state of some

control qubits. For example, the controlled-controlled-not (CC-NOT) gate shown in Figure 1(b), with two control qubits c_0 and c_1 performs the following transformation

$$|c_0, c_1, x\rangle \xrightarrow{\boldsymbol{CC-NOT}} |c_0, c_1, (c_0.\bar{c_1} \oplus x)\rangle \tag{7}$$

Thus, it inverts x when $c_0 = 1$ and $c_1 = 0$. We use the notational convention $|c_0, c_1, x\rangle = |c_0\rangle\,|c_1\rangle\,|x\rangle$. If a gate becomes active when a control qubit is 1, then that is indicated by a solid circle, (for c_0) and if a gate becomes active when a control qubit is 0 then that is indicated by an open circle, (for c_1). The CC-NOT gate transforms the basis vectors $|100\rangle$ and $|101\rangle$ to $|101\rangle$ and $|100\rangle$ respectively and leaves all the remaining six basis vectors unchanged. The qubit affected by the operation of a controlled gate is called the target qubit. If we initialize x to 0, then this gate can be viewed as a *quantum comparator* which sets the target qubit to $|1\rangle$ when $c_0 > c_1$ and leaves it unchanged otherwise.

2.3 Quantum Copy and Switch Gates

The simplest controlled gate is the controlled-not (CNOT) gate shown in Figure 1(c). The function of this gate is given by the mapping:

$$|s\rangle\,|t\rangle \xrightarrow{CNOT} |s\rangle\,|s \oplus t\rangle \tag{8}$$

Hence bit t is inverted when $s = 1$ and remains unchanged when $s = 0$. This gate functions as a NOT gate for the lower or target qubit when the control qubit is in state $|1\rangle$. When $t = 0$ we see that the mapping is of the form $|s\rangle\,|0\rangle \xrightarrow{CNOT} |s\rangle\,|s\rangle$, thus a CNOT gate can also be viewed as a copier which copies the upper or source qubit on to the lower or target qubit when the target qubit is initialized to state $|0\rangle$. Note that this copy operation can only be done when the upper qubit is in one of the two basis states: $|0\rangle$ or $|1\rangle$. For a source qubit in the general state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, the mapping is given by:

$$(\alpha\,|0\rangle + \beta\,|1\rangle)_s\,|0\rangle_t \xrightarrow{CNOT} \alpha\,|00\rangle_{st} + \beta\,|11\rangle_{st} \tag{9}$$

The output state is either 00 with probability $|\alpha|^2$ or 11 with probability $|\beta|^2$ and we have copied 0 and 1 bits to the target. We shall use such gates in our concentrator design.

The basic gate used to build quantum switching networks is the controlled swap gate or switch gate, which is shown in Figure 1(d). A switch gate is a multi-qubit gate which swaps two sets of qubits or quantum packets when a control qubit c is $|1\rangle$ and passes them through unchanged, otherwise [14], [15]. These two states of the switch gate are referred to as the cross

and through states respectively. Thus, this gate can be used as a reconfigurable 2×2 switch to route quantum bit packets. The function of this gate can be represented as

$$|\boldsymbol{x}\rangle\,|\boldsymbol{y}\rangle\,|0\rangle_c \xrightarrow[Through]{Switch} |\boldsymbol{x}\rangle\,|\boldsymbol{y}\rangle\,|0\rangle_c \tag{10}$$

$$|\boldsymbol{x}\rangle\,|\boldsymbol{y}\rangle\,|1\rangle_c \xrightarrow[Cross]{Switch} |\boldsymbol{y}\rangle\,|\boldsymbol{x}\rangle\,|1\rangle_c \tag{11}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are equal length binary strings. The thick lines in Figure 1(d) for $\boldsymbol{x}$ and $\boldsymbol{y}$ indicate that there are multiple qubits on them. An implementation of the switch gate with strings $\boldsymbol{x}$ and $\boldsymbol{y}$ of length 1, using two controlled-not (CNOT) and one CC-NOT gate is shown in Figure 1(e). If $\boldsymbol{x} = x_1x_0$ and $\boldsymbol{y} = y_1y_0$ are strings of length two, then the transformation done by the switch gate is given by:

$$|x_1, x_0\rangle\,|y_1, y_0\rangle\,|0\rangle_c \xrightarrow[Through]{Switch} |x_1, x_0\rangle\,|y_1, y_0\rangle\,|0\rangle_c \tag{12}$$

$$|x_1, x_0\rangle\,|y_1, y_0\rangle\,|1\rangle_c \xrightarrow[Cross]{Switch} |y_1y_0\rangle\,|x_1, x_0\rangle\,|1\rangle_c \tag{13}$$

3 Classical Sparse Crossbar Concentrators

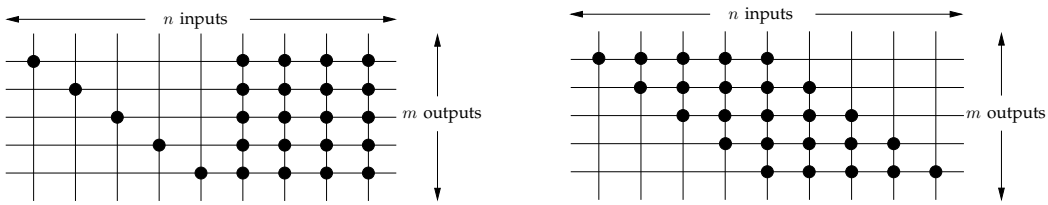
An (n, m)-sparse crossbar network is a matrix of crosspoints or switches with m rows and n columns. Each crosspoint acts as a simple 2×2 switch which can either swap the data on its two inputs onto its outputs or pass them through unchanged. We refer to these two states of the crosspoint as the "cross" state and the "through" state respectively.

An (n, m)-sparse crossbar concentrator is a (n, m)-sparse crossbar in which any k inputs, $k \leq m$, can be routed over nonintersecting paths to some k outputs. Any sparse crossbar is a concentrator if its crosspoint distribution is such that the constraints of Hall's theorem are satisfied. This theorem is stated below:

Theorem 1 (Hall's Theorem)**.** *Let O be a finite set and let $Y_1, Y_2, \ldots, Y_r$ be arbitrary subsets of O. There exist distinct elements $y_i \in Y_i$, $1 \leq i \leq r$ if and only if the union of any k of $Y_1, Y_2, \ldots, Y_r$ contains at least k elements.*

Let the set O in the theorem denote the set of outputs of a sparse crossbar and $Y_1, Y_2, \ldots, Y_r$ represent the neighbor sets of some r inputs $s_1, s_2, \ldots, s_r$ respectively, i.e., Y_i is the subset of outputs in O to which input s_i can be connected, $1 \leq i \leq r$. Then if the union of $Y_1, Y_2, \ldots, Y_r$ contains at least r outputs for any choices of $s_1, s_2, \ldots, s_r$ in the input set, and any r, $1 \leq r \leq m$, then Hall's theorem implies that the sparse crossbar is a concentrator.

Nakamura and Masson in [28] gave a lower bound of $m(n - m + 1)$ on the crosspoint complexity, i.e., number of crosspoints, for (n, m)-sparse crossbar concentrators by showing that each output needs to share crosspoints with at least $n - m + 1$ inputs. Oruç et al in [24]



(a) Fat-slim crossbar concentrator. (b) Banded crossbar concentrator.

Fig. 2: Classical sparse crossbar concentrators, n=9, m=5.

and [25] gave explicit crossbar structures of optimal concentrators which achieved this bound for any n and m. They used Hall's theorem to show that certain $n \times m$ sparse crossbar structures with exactly $m(n-m+1)$ crosspoints can act as concentrators.

Two such optimal concentrators, the fat-slim and banded sparse crossbar concentrators, which we shall be using extensively, are shown in Figure 2. As seen in Figure 2(a), in a fat-slim crossbar concentrator where, unlike in the original construction, we placed the slim part on the left, the input columns can be divided into a fat portion of $n-m$ inputs in which each input is connected to all outputs and a slim portion of m inputs in which each input is connected to a distinct output. The crosspoints in the slim part do not need to fall on a diagonal, any one-to-one connection between the m inputs and m outputs would work. In a banded crossbar concentrator (Figure 2(b)) all the crosspoints form a transverse band in the middle. Note that in these sparse crossbars each of the m outputs is connected to $n-m+1$ inputs.

4 Quantum Packets and Concentration Assignments

A quantum packet consists of a set of *data qubits* and one additional qubit which we refer to as the *routing qubit*. We assume that quantum packets composed of qubits are routed over a quantum concentrator. Reversibility considerations in quantum systems mean that unlike classical systems no connecting wire or input/output line can remain empty. We use the routing qubit to overcome this constraint, the routing qubit is set to $|1\rangle$ to indicate the presence of a quantum packet and to $|0\rangle$ to indicate an empty wire or absence of a packet. Whenever we write $|a\rangle$, where a is a binary variable, then this represents the state of a quantum bit. The binary variable itself is indicated as a. A quantum packet is defined as [29]:

Definition 1 (Quantum packet)**.** Let an input have m n_d-bit packets, $d_1, \ldots, d_m$. If packet d_i is to be concentrated with probability p_i, then the source at this input feeds into the concentrator a *quantum packet* of the following form:

$$\sum_{i=1}^{m} \alpha_i \, |r_i, d_i\rangle \tag{14}$$

where $|\alpha_i|^2 = p_i$ and $r_i = 1$. We refer to the individual components of the quantum packet, the

bit strings (r_i, d_i) as *classical packets*. The r_i denote the routing bits in the classical packets. The length of the quantum packet is $n_d + 1$ qubits. We refer to these strings as classical packets as they represent a basis state (with no superposition) of the constituent qubits and any group of quantum bits in a basis state are conceptually equivalent to a group of classical bits having the same value.

If the input source has no packets to concentrate then the empty line is indicated by a single $n_d + 1$-bit string in which the routing bit is set to 0, and the data bits can be set to any arbitrary value.

An input *concentration pattern* for an n-input concentrator is a sequence of classical packets, each of which belongs to a quantum packet on the n inputs from top to bottom. A *quantum concentration assignment* is a superposition of a set of concentration patterns. We define these terms formally as follows:

Definition 2 (Quantum concentration assignment)**.** A *quantum concentration assignment* for an n-input concentrator is a superposition of a set of t concentration patterns of the form:

$$\sum_{j=1}^{t} \gamma_j \left|(r_{j1}, d_{j1}), \cdots, (r_{jn}, d_{jn})\right\rangle \tag{15}$$

where the *concentration pattern* $|(r_{j1}, d_{j1}), \cdots, (r_{jn}, d_{jn})\rangle$ consists of n classical packets in which (r_{ji}, d_{ji}) is the jth classical packet at input i, and $|\gamma_j|^2$ is the probability of the jth concentration pattern being realized with $\sum_{j=1}^{t} |\gamma_j|^2 = 1$.

If the quantum packets at all the inputs are independent and of the kind given in Eqn. (14) then the quantum assignment can be expressed as a tensor product of the quantum packets as follows:

$$\bigotimes_{i=1}^{n} |Q_i\rangle = \bigotimes_{i=1}^{n} \left(\sum_{j=1}^{t_i} \alpha_{ij} \left|r_{ij}, d_{ij}\right\rangle \right) \tag{16}$$

where $|Q_i\rangle = \sum_{j=1}^{t_i} \alpha_{ij} \left|r_{ij}, d_{ij}\right\rangle$ is the quantum packet on input i. The tensor product in Eqn. (16) expanded to a quantum concentration assignment of the form in Eqn. (15) contains $\prod_{i=1}^{n} t_i$ concentration patterns.

As an example, consider three inputs indexed by $1, 2$ and 3 having the quantum packets: $\frac{1}{\sqrt{2}}(|1, d_{11}\rangle + |1, d_{12}\rangle)$, $|1, d_{21}\rangle$ and $\frac{1}{\sqrt{3}} |1, d_{31}\rangle + \frac{\sqrt{2}}{\sqrt{3}} |1, d_{32}\rangle$ respectively. Then the quantum concentration assignment is given by:

$$\begin{aligned} & \frac{1}{\sqrt{2}} (|1, d_{11}\rangle + |1, d_{12}\rangle) \otimes |1, d_{21}\rangle \otimes \left(\frac{1}{\sqrt{3}} |1, d_{31}\rangle + \frac{\sqrt{2}}{\sqrt{3}} |1, d_{32}\rangle \right) \\ = \; & \frac{1}{\sqrt{6}} |(1, d_{11}), (1, d_{21}), (1, d_{31})\rangle + \frac{1}{\sqrt{3}} |(1, d_{11}), (1, d_{21}), (1, d_{32})\rangle \\ & + \frac{1}{\sqrt{6}} |(1, d_{12}), (1, d_{21}), (1, d_{31})\rangle + \frac{1}{\sqrt{3}} |(1, d_{12}), (1, d_{21}), (1, d_{32})\rangle \end{aligned} \tag{17}$$

Thus the quantum concentration assignment consists of a superposition of four concentration patterns, two of which have a probability $1/3$ and the other two a probability $1/6$ of being observed on measurement. A quantum concentrator can route such patterns contained in the input quantum concentration assignment in parallel.

5 Quantum Sparse Crossbar Concentrators

An input concentration pattern for a concentrator is said to be *capacity achieving* if no greater than m packets in the pattern have their routing bits equal to 1, where m is the number of outputs of the concentrator. We call a quantum concentration assignment *capacity achieving* if all of its concentration patterns are capacity achieving. Also, we refer to packets with routing bit set to 1 as valid packets.

A *quantum sparse crossbar* is obtained from a classical crossbar by replacing the classical crosspoints by quantum crosspoints which can switch quantum packets. A quantum crosspoint can be viewed as a configurable 2×2 switch which either swaps or passes through unchanged to its outputs the two quantum packets incident at its inputs. Reversibility in quantum systems implies that, for a quantum sparse crossbar, unlike a classical sparse crossbar, each crosspoint needs to have qubits coming in on each of its two inputs and qubits leaving on each of its outputs. As mentioned earlier in Section 4, an absence of a packet or an empty wire is indicated by a quantum bit string with the routing bit set to 0. Thus, in the quantum domain, the m empty wires coming in from the left in an $n \times m$ sparse crossbar can be represented by blocks of quantum bits in which the routing bit is set to 0. We can imagine m additional packet sources at these wires which generate quantum bit blocks in which the routing bit is always set to 0. The same reasoning can be applied to the n empty wires leaving the sparse crossbar at the bottom. This means that they can be viewed as n additional outputs and if the sparse crossbar is a concentrator then the exiting quantum bit strings on these outputs have their routing bits equal to 0 as long as the input pattern is capacity achieving. This effectively creates a "square" $(n+m) \times (n+m)$ crossbar network from a "rectangular" $n \times m$ network in which only the n inputs (on the top) can get valid packets. If the sparse crossbar is a concentrator and the input pattern is capacity achieving then all the valid packets are concentrated to the m outputs on the right hand side. If we number the inputs on the top 1 to n from left to right, the inputs on the left $n+1$ to $n+m$ from top to bottom, the outputs on the right 1 to m and outputs on the bottom $m+1$ to $m+n$ as shown in Figure 3 then:

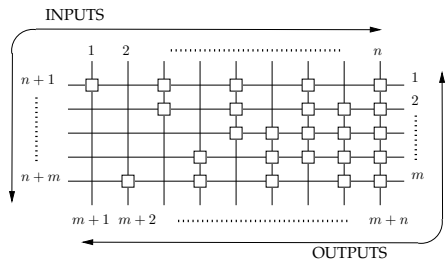


Fig. 3: Numbering of inputs and outputs in an $n \times m$ quantum sparse crossbar.

Definition 3. An $n \times m$ quantum sparse crossbar ($n \geq m$) is called an (n, m)-*quantum sparse crossbar concentrator* and denoted QSC(n, m) if, given any capacity achieving input pattern with k valid packets, where $k \leq m$, the k packets can be routed to some k of the first m outputs. That is, for a capacity achieving input concentration pattern $|P\rangle = |(r_1, d_1) \cdots (r_n, d_n)\rangle$, a pattern $|(0, d_{n+1}) \cdots (0, d_{n+m})\rangle$ of m packets, each with routing qubit set to $|0\rangle$ and a set of auxiliary qubits initialized to state $|0\rangle$ the following transformation occurs:

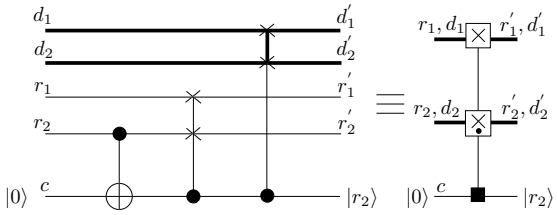
$$|\underbrace{(r_1, d_1) \cdots (r_n, d_n)}_{|P\rangle\text{: from top}}\rangle|\underbrace{(0, d_{n+1}) \cdots (0, d_{n+m})}_{\text{from left}}\rangle\, |00 \cdots 0\rangle_{aux} \xrightarrow{QSC(n,m)}$$

$$|\underbrace{(r'_1, d'_1) \cdots (r'_m, d'_m)}_{\text{on right}}\rangle|\underbrace{(0, d'_{m+1}) \cdots (0, d'_{n+m})}_{\text{bottom}}\rangle\, |\Psi_P\rangle_{aux} \quad (18)$$

where if $\mathcal{R}_1 = \{|r_i, d_i\rangle \ \forall \ r_i = 1\}, \ 1 \leq i \leq n$, is the set of valid input packets and $\mathcal{R}_2 = \{|r'_j, d'_j\rangle \ \forall \ r'_j = 1\}, \ 1 \leq j \leq m$, is the set of valid packets at the outputs then $\mathcal{R}_2 = \mathcal{R}_1$. Here $|00 \cdots 0\rangle_{aux}$ represents the state of a set of auxiliary qubits, one per each crosspoint, and all of which are in state $|0\rangle$.

The auxiliary qubits ensure reversible operation as their final state, $|\Psi_P\rangle$, encodes the state of the crossbar, i.e., the states of the internal crosspoints or equivalently the input-output mapping. In a quantum sparse crossbar concentrator, a subset of inputs can potentially be matched to more than one set of outputs. Even if an input set can be matched with only one output set, it is possible that several one-to-one maps realize this matching. Therefore, a routing algorithm is needed to fix an output set and an input-output mapping for a capacity achieving input pattern. We describe such an algorithm in the next section.

If a quantum crosspoint is configured by using only the information contained in the quantum packets incident on its inputs, then we call such a quantum crosspoint a self-routing crosspoint. A self-routing QSC(n, m) is a sparse crossbar built using self-routing quantum crosspoints that can realize all maps of the kind given in Eqn. (18).



Inputs		Outputs				
r_2	r_1	r_2'	r_1'	d_2'	d_1'	State
0	0	0	0	d_2	d_1	Through
0	1	0	1	d_2	d_1	Through
1	0	0	1	d_1	d_2	Cross
1	1	1	1	d_1	d_2	Cross

Fig. 4: Circuit for the quantum crosspoint. Fig. 5: Input-output map for a quantum crosspoint.

5.1 Self-Routing Quantum Crosspoints

The crosspoints in the classical sparse crossbar are replaced by quantum crosspoints for which the circuit and corresponding schematic representation is given in Figure 4. Here the upper input with packet $|r_1,\ d_1\rangle$ corresponds to the input line incident on a crosspoint from the top and the lower input with packet $|r_2,\ d_2\rangle$ corresponds to the input line coming in from the left. The data bit strings d_1, d_2 and the routing bits r_1, r_2 are shown separately for clarity. The thick lines indicate that d_1 and d_2 are bit strings with multiple bits. Each crosspoint uses an auxiliary control qubit initialized to state $|0\rangle$, which is then set according to the map in Figure 5 and used to control the setting ("through" or "cross") of the switch gate. When the control qubit is set to state $|1\rangle$, the input packets get swapped and when the control qubit is in state $|0\rangle$, the input packets go through unchanged. In the crosspoint circuit the CNOT gate functions as a copier which sets the state of the control qubit to $|r_2\rangle$. This qubit is then used to control the two swap gates which switch r_1, r_2 and d_1, d_2 respectively. So, the input-output mapping performed by the quantum crosspoint can be represented as:

$$\begin{aligned} |r_1, d_1, 0, d_2\rangle\,|0\rangle &\xrightarrow[Through]{Crosspoint} |r_1, d_1, 0, d_2\rangle\,|0\rangle \\ |r_1, d_1, 1, d_2\rangle\,|0\rangle &\xrightarrow[Cross]{Crosspoint} |1, d_2, r_1, d_1\rangle\,|1\rangle \end{aligned} \tag{19}$$

We can see that the auxiliary control qubit is always set to state $|r_2\rangle$, i.e., the packets are swapped when the routing qubit of the packet coming in from the left (r_2) is in state $|1\rangle$ and go through unchanged when this qubit is in state $|0\rangle$. In the schematic representation for the quantum crosspoint in Figure 4, the filled circle or dot in the second box signifies that the routing qubit of the packet coming in from the left is used to set the control qubit. Although the switch control is determined fully by just r_2, we cannot use its corresponding qubit by itself (without the auxiliary control qubit) to set the switch. This is because the qubit for r_2 is a part of the packet incident on the lower input and itself needs to be switched along with that packet, so although this qubit can act as a control qubit for all the rest of the qubits in the packet, it cannot

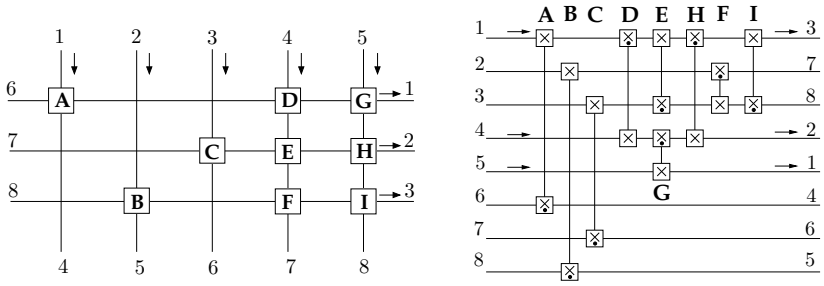


Fig. 6: Self-routing fat-slim QSC(5,3): (a) Crossbar structure (b) Example for self-routing: Inputs 1, 4 and 5 are concentrated.

be a control qubit for switching itself. Also, we are not using r_1 in the switch to determine the routing state, but this bit is still relevant as eventually at the output of the concentrator valid packets are identified by examining the routing bit values. Another reason to consider r_1 is the fact that crosspoint switches are interconnected to form the crossbar, and the packet from the upper input at one crosspoint switch may be incident at the lower input of a switch in later stages. In this scenario the qubit corresponding to r_1 would be the auxiliary control for this later stage switch. As the state of the quantum crosspoint is determined fully by just using the information about the routing bits from the two input packets, the paths in the sparse crossbar are determined in a self-routing fashion.

5.2 The Self-Routing Scheme

Starting with input 1, the routing of packets proceeds from top to bottom in a column for an input and then to next higher numbered input in the next column. The control qubit is not restored to its original state immediately and we give a method to restore the control qubit at the output of the concentrator in Section 8.

Unlike in the classical case, all n inputs (plus the m additional inputs on the left) have quantum bit strings incident on them. We distinguish the subset of inputs having packets for concentration by setting the routing qubits in the headers of packets at these inputs to 1. A self-routing quantum concentrator derived from a variant of the classical fat-slim $(5, 3)$-sparse crossbar concentrator is shown in Figure 6. The square boxes in Figure 6(a) indicated by letters A–I are quantum crosspoints. In this concentrator, valid packets (packets with routing bit 1) come only on inputs 1–5, and if the input pattern is capacity achieving, they exit only on outputs 1–3. These inputs and outputs are indicated by arrows in the figure. In the process of self-routing, the crosspoints are traversed from top to bottom in a column and from left to right in a row.

The crossbar in Figure 6(a) is redrawn in Figure 6(b) using the schematic representation of quantum crosspoints shown in Figure 4. We have not shown the auxiliary control qubits to maintain clarity. The circuit in Figure 6(b) is obtained by putting all crosspoints in the through state, i.e., input line at a crosspoint coming in from the top is "bent" and connected to output going out to the right and input line coming in from the left is "bent" and connected to the output going out to the bottom, and following the sequence of crosspoints encountered from a crossbar input to a crossbar output. For example, with all crosspoints in through state the path from input 2 goes through crosspoints B and F and comes out at output 7. At crosspoint B input 2 and input 8 interact, with input 8 coming in from the left, hence in Figure 6(b) crosspoint B connects input 2 and input 8 with the box with the filled circle or dot connected to input 8. Similarly at crosspoint F, the left input lies on the path from input 2 while the top input lies on the path from input 3 (via crosspoints C and E). Hence crosspoint F connects lines for inputs 2 and 3 with the dotted box connected to the line for input 2. The assignments of dotted boxes between the inputs of the quantum crosspoints in Figure 6(b) are designed to facilitate the self-routing algorithm as illustrated by an example below:

Consider an input concentration pattern with valid packets at inputs 1, 4 and 5 for the crossbar shown in Figure 6(b), i.e., only packets appearing at inputs 1, 4 and 5 have routing bits set to 1 (indicated by arrows). At all the other inputs, the incident packets have routing bits set to 0. At crosspoint A the upper input, i.e., input 1 has a valid packet and the lower input has a packet with routing bit set to 0. Thus, this situation corresponds to the $r_1 = 1, r_2 = 0$ case in Figure 5 and A is set to pass the packets through unchanged. Proceeding to the next stage, we see that at crosspoint D, the routing bit of the packets at both the lower and upper inputs is 1, hence D swaps its input packets onto its outputs. Continuing in this way we see that the packet from input 1 takes the path A→D→G to output 1. Similarly the packets from input 4 and input 5 take the paths D→E→H and G→H→I to outputs 2 and 3 respectively. We note that it is not required that all qubits travel through the same number of quantum gates in a quantum circuit as it also happens in this quantum circuit implementation of a fat-and-slim crossbar.

The intuition behind the routing scheme is as follows: In the circuit shown in Figure 6(b) routing priority is given to the packet incident on the crosspoint input connected to the "dotted" box of a quantum crosspoint, this is equivalent to routing according to the packet coming in from the left in the crossbar. If the routing bit of the packet from the left is 1 then switch is set in a cross state, and this is equivalent to passing a valid packet coming in from the left to the

right irrespective of the packet incident from the top. Thus, once a valid packet is routed from the top to the right, it goes through unimpeded to the end of the row. In other words, once an input is matched to an output this decision remains unchanged for the subsequent steps of the routing process. The crosspoint is set in a through state when the packet at the input to the "dotted" box has its routing bit equal to 0, and this corresponds to a turn from left to the bottom in the crossbar. This observation combined with the previous argument means that, in a column of the crossbar, a packet gets passed from top to bottom from one row to the next until it encounters a crosspoint where the packet from the left has a routing bit set to 0.

This self-routing scheme can be used for any sparse crossbar structure, it is not limited to concentrators. An interesting question to ask is whether all known optimal sparse crossbar concentrator structures allow concentration when a self-routing scheme of the form described above is used to route packets. In the subsequent sections we prove that while this is not true for all optimal sparse crossbar concentrators, it holds for fat-slim and banded crossbars.

6 Self-Routing on Quantum Sparse Crossbar Concentrators

We prove that the self-routing scheme described in Section 5.1 concentrates packets up to the output capacity of the concentrator, for two families of sparse crossbar structures, namely the fat-slim and banded crossbars. All the proofs are given for a quantum input assignment consisting of a single concentration pattern. This is sufficient as the linearity of quantum systems implies that, for any general input quantum concentration assignment, these results apply to every concentration pattern contained in its superposition and hence to the entire input assignment.

We first introduce some notation that will be used throughout the rest of the paper. For a quantum (n, m)-sparse crossbar ($n \geq m$) in which packets are routed using our self-routing scheme, we use the following notation:

1) The set of inputs is denoted by $\mathcal{I} = \{1, 2, \ldots, n\}$ and the output set by $\mathcal{O} = \{1, 2, \ldots, m\}$ where $1 \leq m \leq n$.
2) The $n \times m$ adjacency matrix, $\boldsymbol{A}$, is given by:

$$\boldsymbol{A} = \{a_{ij}\} = \begin{cases} 0, & \text{no crosspoint between input } i \text{ and output } j, \\ 1, & \text{crosspoint between input } i \text{ and output } j. \end{cases}$$

3) $\mathcal{A}_i = \{j : \text{for all } a_{ij} = 1, 1 \leq i \leq n\}$, is the neighbor set for input i, i.e., the set of outputs which can be connected to input i.

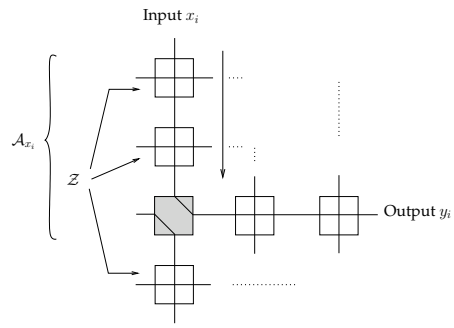


Fig. 7: Output matching y_i for input x_i.

4) $\mathcal{X}_1^r = \{x_1, x_2, \ldots, x_r\}$ is an ordered set of r distinct inputs, i.e., $x_i \in \mathcal{I},\ 1 \leq i \leq r$, and $x_1 < x_2 < \cdots < x_r$ for all $r = 1, \ldots, m$.
5) $\mathcal{Y}_1^r = \{y_1, y_2, \ldots, y_r\}$ is the set of outputs to which the inputs in $\mathcal{X}_1^r$ are matched using self routing with output y_i being matched to input x_i, where, $y_i \in \mathcal{A}_{x_i}$, $i = 1, \ldots, r$. If input x_i can not be matched to any output then $y_i = \emptyset$, the empty set.
6) We denote subsets of $\mathcal{Y}_1^r$ as follows: $\mathcal{Y}_0 = \emptyset$, the empty set, $\mathcal{Y}_1 = \{y_1\}$, $\mathcal{Y}_1^b = \{y_1, y_2, \ldots, y_b\}$, and $\mathcal{Y}_a^b = \{y_a, y_{a+1}, \ldots, y_b\},\ \forall\ a \leq b \leq r$.

We now show that using the self-routing scheme described above in Section 5.2, certain families of sparse crossbar concentrators, namely fat-slim and banded crossbar concentrators can correctly route any capacity achieving input pattern.

Lemma 1. *Let $\mathcal{Z} \subseteq \mathcal{O}$ be the subset of outputs to which packets have already been matched, using self-routing, before routing begins on input x_i, $1 \leq i \leq r$. Then, the output y_i, to which input x_i is matched is given by $y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap \mathcal{Z}'\}$, where $\mathcal{Z}' = \mathcal{O} \setminus \mathcal{Z}$ is the complement of the set $\mathcal{Z}$.*

Proof: All outputs which have been concentrated to (matched) before routing the packet on input x_i, (i.e., in $\mathcal{Z}$) correspond to rows which have a packet with routing bit $r = 1$ coming from the left on them. We know from the table for the auxiliary control qubit (Figure 5) that all crosspoints in rows corresponding to set $\mathcal{Z}$ will be set to the "cross" state, see Figure 7. This means that only a crosspoint from the set of rows corresponding to $\mathcal{Z}'$ will be set to the "through" state. This in turn means $y_i \in \mathcal{Z}'$. Also, obviously, $y_i \in \mathcal{A}_{x_i}$. Thus, $y_i \in \mathcal{A}_{x_i} \cap \mathcal{Z}'$. Since the routing process proceeds from the top to bottom, i.e., in increasing order of row/output number, the lowest numbered available output is matched. Hence, $y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap \mathcal{Z}'\}$. $\square$

Lemma 2. *Let y_i be the output to which input x_i is matched, $1 \leq i \leq r$. Then,*

$$y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_1^{i-1})'\}$$

Proof: As the routing of packets at inputs is initiated in increasing order of input index from left to right, the packets at inputs $x_1, \ldots, x_{i-1}$ are routed before routing starts on input x_i. Thus, the set of outputs matched before routing starts on input x_i is $\mathcal{Y}_1^{i-1}$. Substituting $\mathcal{Z} = \mathcal{Y}_1^{i-1}$ in Lemma 1 we get the desired result. □

Lemma 2 essentially asserts that, at any input, the first or lowest numbered unmatched output is selected as a match for that input during self-routing. If an input x_i cannot be matched to any output in its neighbor set $\mathcal{A}_{x_i}$, then $y_i = \emptyset$.

Lemma 3. *If $i \neq j$ and $y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_1^{i-1})'\} \neq \emptyset$ and $y_j = \min_z \{z \in \mathcal{A}_{x_j} \cap (\mathcal{Y}_1^{j-1})'\} \neq \emptyset$. Then, $y_i \neq y_j$, $1 \leq i, j \leq r \leq m$.*

Proof: Without loss of generality, assume $i < j$, then $y_i \in \{y_1, \ldots, y_{j-1}\} = \mathcal{Y}_1^{j-1}$. Also $y_j = \min_z\{z \in \mathcal{A}_{x_j} \cap (\mathcal{Y}_1^{j-1})'\}$ implies that $y_j \in (\mathcal{Y}_1^{j-1})'$. Hence, $y_i \neq y_j$. □

Theorem 2. *An (n, m)-sparse crossbar network is a self-routing concentrator if*

$$y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_1^{i-1})'\} \neq \emptyset$$

for all $i = 1, \ldots, r$, and for every r-input ordered subset $\mathcal{X}_1^r = \{x_1, \ldots, x_r\}$, $1 \leq r \leq m$.

Proof: Let $\mathcal{Y}_1^r = \{y_1, \ldots, y_r\}$ where $y_i = \min_z \{z \in \mathcal{A}_{x_i} \cap (\mathcal{Y}_1^{i-1})'\} \neq \emptyset$ for all $i = 1, \ldots, r$. By Lemma 3 all elements of $\mathcal{Y}_1^r$ are distinct and form a matching for inputs in $\mathcal{X}_1^r$. By Lemma 2, $\mathcal{Y}_1^r$ corresponds to the set of outputs chosen by self-routing packets on the (n, m)-sparse crossbar. If such an output matching exists for every ordered r input subset, $1 \leq r \leq m$ then this implies that any r inputs can be connected to r distinct outputs using self-routing. Hence, such an (n, m)-sparse crossbar network is a self-routing concentrator. □

A quantum concentrator derived by replacing classical crosspoints by quantum crosspoints in a classical sparse crossbar concentrator is not always self-routable by the algorithm described above. One such scenario is shown in Figure 8. For the sparse crossbar in this figure the union of any k columns (or equivalently input neighbor sets) contains crosspoints in least k distinct rows (or outputs) for all $k = 1, \ldots, 5$, thus by Hall's theorem it is a $(9, 5)$-concentrator. The set of inputs with packets to be concentrated is given by $\mathcal{X}_1^5 = \{1, 3, 5, 6, 7\}$. The output to which an input is matched by following the self-routing scheme is indicated by the shaded crosspoints, therefore, input 1 is matched to output 1, input 3 to output 2 and so on. The crossed out crosspoints indicate the outputs to which an input is not matched during self-routing. The matched outputs for the first four inputs in $\mathcal{X}_1^5$ form the set $\mathcal{Y}_1^4 = \{1, 2, 3, 4\}$. Thus for input 7 (which is the fifth input in $\mathcal{X}_1^5$) we get $y_5 = \min_z \{z \in \mathcal{A}_7 \cap (\mathcal{Y}_1^4)'\} = \min_z\{z \in \{1, 2, 3, 4\} \cap \{5\}\} = \emptyset$. Hence, this concentrator can not self-route all input subsets using the algorithm we described. This example

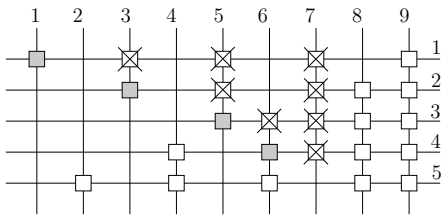


Fig. 8: Conflict in self-routing.

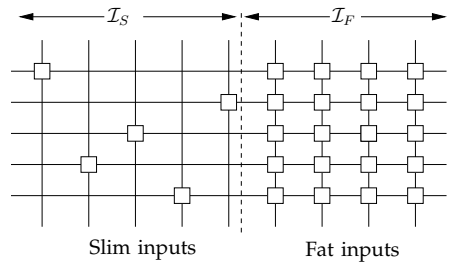


Fig. 9: Partitions of fat-slim crossbar.

shows that not all sparse-crossbar concentrators are self-routable using our algorithm.

We now show that fat-slim and banded sparse crossbar concentrators can self-route by showing that for these concentrators Theorem 2 is always satisfied. As part of the proofs we give explicit expressions for the input-output mapping realized for concentration on these structures.

Self-Routing Fat-Slim QSC(n, m)

Definition 4 (Fat-Slim Crossbar)**.** An (n, m)-sparse crossbar network is called fat-slim if we can partition the input set $\mathcal{I}$ into two subsets: $\mathcal{I}_S$ (slim inputs) and $\mathcal{I}_F$ (fat inputs) as shown in Figure 9 with neighbor sets for input i described as follows:

$$i \in \mathcal{I}_S \quad \Leftrightarrow \quad 1 \leq i \leq m;\ \mathcal{A}_i = \pi(i) \tag{20a}$$

$$i \in \mathcal{I}_F \quad \Leftrightarrow \quad m < i \leq n;\ \mathcal{A}_i = \{1, \ldots, m\} = \mathcal{O} \tag{20b}$$

where π is a permutation on the elements of the set $\{1, \ldots, m\}$.

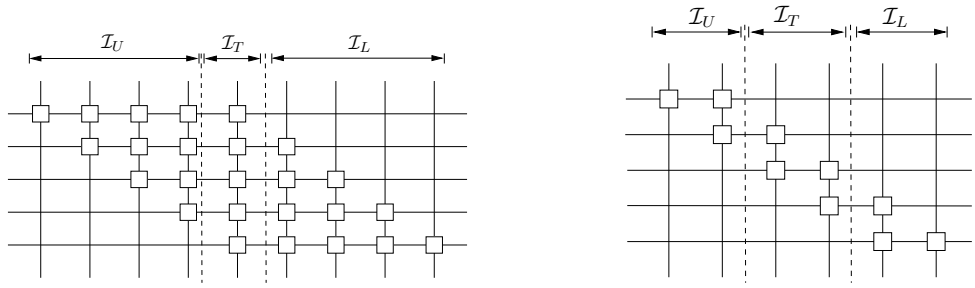
Every fat-slim (n, m)-sparse crossbar is an optimal (n, m)-concentrator with $m(n - m + 1)$ crosspoints [24]. We now show that any capacity achieving input pattern can be self-routed on a fat-slim sparse crossbar concentrator.

Theorem 3. *For the fat-slim QSC(n, m) let $\mathcal{X} = \{x_1, x_2, \ldots, x_r\}$ be any ordered r-input subset where $x_1 < x_2 < \cdots < x_r$, $\forall r, 1 \leq r \leq m$. Then, there exists an output matching, $\mathcal{Y} = \{y_1, y_2, \ldots, y_r\}$ for $\mathcal{X}$ obtained as result of self-routing the fat-slim QSC(n, m) and it is given by*

$$y_i = \begin{cases} \pi(x_i), & x_i \in \mathcal{I}_S, \\ b_{i-(m-a)}, & x_i \in \mathcal{I}_F. \end{cases}$$

where $\mathcal{B} = \{b_1, \ldots, b_a\} = (\{\pi(x_i) \in \mathcal{I}_S\})'$ such that $b_1 < b_2 < \cdots < b_a$, $a = |\mathcal{B}|$, $i = 1, 2, \ldots, r$.

Proof: The proof is inductive in nature and given in Appendix A. □



(a) Partitions of banded crossbar, $n = 9, m = 5$, $(n \geq 2m-1)$.

(b) Partitions of banded crossbar, $n = 6, m = 5$, $(m \leq n < 2m-1)$.

Fig. 10: Partitions of banded sparse crossbar concentrator.

Self-Routing Banded QSC(n, m)

We now show that for banded sparse crossbar concentrators our self-routing scheme can find an r-output matching for any r input subset ($r \leq m$).

Definition 5 (Banded Crossbar)**.** An (n, m)-sparse crossbar is called banded if the set of inputs, $\mathcal{I}$, can be partitioned into three sets $\mathcal{I}_U, \mathcal{I}_T$ and $\mathcal{I}_L$ as shown in Figure 10 with the corresponding neighbor sets for the inputs as follows:

If $n \geq 2m-1$ then:

$$i \in \mathcal{I}_U \quad \Leftrightarrow \quad 1 \leq i \leq m-1;\ \mathcal{A}_i = \{1, 2, \ldots, i\} \tag{21a}$$

$$i \in \mathcal{I}_T \quad \Leftrightarrow \quad m \leq i \leq n-m+1;\ \mathcal{A}_i = \{1, 2, \ldots, m\} = \mathcal{O} \tag{21b}$$

$$i \in \mathcal{I}_L \quad \Leftrightarrow \quad n-m+2 \leq i \leq n;\ \mathcal{A}_i = \{i-n+m, \ldots, m\} \tag{21c}$$

If $m \leq n < 2m-1$ then:

$$i \in \mathcal{I}_U \quad \Leftrightarrow \quad 1 \leq i \leq n-m+1;\ \mathcal{A}_i = \{1, 2, \ldots i\} \tag{22a}$$

$$i \in \mathcal{I}_T \quad \Leftrightarrow \quad n-m+2 \leq i \leq m-1;\ \mathcal{A}_i = \{i-n+m, \ldots, i\} \tag{22b}$$

$$i \in \mathcal{I}_L \quad \Leftrightarrow \quad m \leq i \leq n;\ \mathcal{A}_i = \{i-n+m, \ldots, m\} \tag{22c}$$

Note that for $n = 2m-2,\ n-m+2 = m > m-1$. Hence in this case, from Eqn. (22b), $\mathcal{I}_T$ does not exist, but this does not affect the proof below. Also, Eqns. (21a)-(22c) can be written more succinctly as:

$$\mathcal{A}_i = \{\max(1,\ i-n+m), \ldots, \min(i,\ m)\}, \quad i = 1, \ldots, n \tag{23}$$

Every banded (n, m)-sparse crossbar is an optimal (n, m)-concentrator with $m(n-m+1)$ cross-points [25].

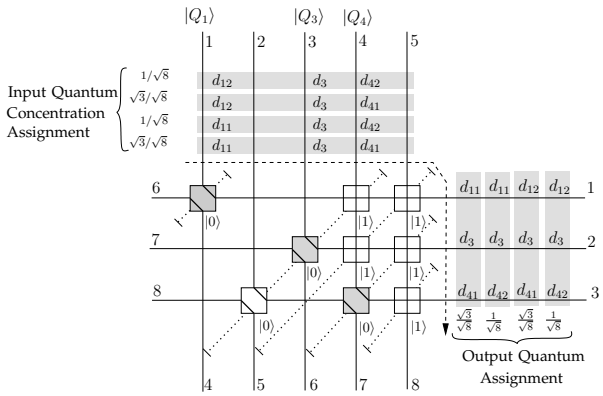


Fig. 11: Self-Routing on fat-slim QSC(5,3).

Theorem 4. *For the banded QSC(n,m) let $\mathcal{X} = \{x_1, x_2, \ldots, x_r\}$ be any ordered r-input subset where $x_1 < x_2 < \cdots < x_r$, $\forall\, r, 1 \leq r \leq m$. Then, there exists an output matching, $\mathcal{Y} = \{y_1, y_2, \ldots, y_r\}$ for $\mathcal{X}$ obtained as result of self-routing the banded QSC(n, m) and it is given by*

$$y_i = \max(i,\ x_i - n + m) \quad i = 1, 2, \ldots, r.$$

Proof: The proof follows an inductive approach similar to that used for Theorem 3 and is given in Appendix B. □

7 An Example

We give an example to illustrate self-routing on a fat-slim QSC$(5, 3)$. This concentrator is shown in Figure 11. The quantum packets present at inputs $1, 3$ and 4 are $|Q_1\rangle = \frac{1}{\sqrt{2}}(|1, d_{11}\rangle + |1, d_{12}\rangle)$, $|Q_3\rangle = |1, d_3\rangle$ and $|Q_4\rangle = \frac{\sqrt{3}}{2}|1, d_{41}\rangle + \frac{1}{2}|1, d_{42}\rangle$ respectively. Inputs 2 and 5 do not have any packets. Inputs $6, 7$ and 8 correspond to the three dummy inputs on the left hand side from top to bottom. Thus, in this case, the input quantum concentration assignment is given by

$$\begin{aligned}
& |Q_1\rangle \otimes |0, d_2\rangle \otimes |Q_3\rangle \otimes |Q_4\rangle \otimes |0, d_5\rangle \bigotimes_{i=6}^{8} |0, d_i\rangle \\
= \; & \left(\frac{1}{\sqrt{2}}|1, d_{11}\rangle + \frac{1}{\sqrt{2}}|1, d_{12}\rangle\right) \otimes |0, d_2\rangle \otimes |1, d_3\rangle \otimes \left(\frac{\sqrt{3}}{2}|1, d_{41}\rangle + \frac{1}{2}|1, d_{42}\rangle\right) \\
& \qquad \otimes |0, d_5\rangle \otimes |0, d_6\rangle \otimes |0, d_7\rangle \otimes |0, d_8\rangle \\
= \; & \frac{\sqrt{3}}{2\sqrt{2}}|(1, d_{11}), (0, d_2), (1, d_3), (1, d_{41}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \\
& + \frac{1}{2\sqrt{2}}|(1, d_{11}), (0, d_2), (1, d_3), (1, d_{42}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \\
& + \frac{\sqrt{3}}{2\sqrt{2}}|(1, d_{12}), (0, d_2), (1, d_3), (1, d_{41}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \\
& + \frac{1}{2\sqrt{2}}|(1, d_{12}), (0, d_2), (1, d_3), (1, d_{42}), (0, d_5), (0, d_6), (0, d_7), (0, d_8)\rangle \qquad (24)
\end{aligned}$$

Thus, the input is a superposition of four concentration patterns with co-efficients $\sqrt{3}/\sqrt{8}$, $1/\sqrt{8}, \sqrt{3}/\sqrt{8}$ and $1/\sqrt{8}$ respectively, shown by grey horizontal bars. Since all four patterns are capacity achieving, the quantum assignment is also capacity achieving. The state of the crosspoints is also shown. The shaded crosspoints route the valid packets on inputs $1, 3$ and 4. Measurement at the output will result in one out of the four patterns shown at the output being observed with probabilities $3/8, 1/8, 3/8$ and $1/8$ respectively. Therefore, data packets d_{11} and d_{12} are observed on output 1 with probability $1/2$. Data packet d_3 is observed on output 2 with probability 1 and data packets d_{41} and d_{42} are observed with probability $3/4$ and $1/4$ respectively on output 3. This output state can be explicitly written as:

$$\begin{aligned}
&\frac{\sqrt{3}}{2\sqrt{2}}\,|(1,d_{11}),(1,d_3),(1,d_{41}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \\
&\quad+\frac{1}{2\sqrt{2}}\,|(1,d_{11}),(1,d_3),(1,d_{42}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \\
&\quad+\frac{\sqrt{3}}{2\sqrt{2}}\,|(1,d_{12}),(1,d_3),(1,d_{41}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \\
&\quad+\frac{1}{2\sqrt{2}}\,|(1,d_{12}),(1,d_3),(1,d_{42}),(0,d_6),(0,d_8),(0,d_7),(0,d_2),(0,d_5)\rangle \qquad (25)
\end{aligned}$$

The packets in the concentration patterns are written in increasing order of outputs, with outputs 1–3 on the right and outputs 4–8 located on the bottom. The dashed arrow shows the order in which the crosspoints are traversed during routing, and the crosspoints in one stage are indicated by the dotted diagonals. The initial state of the auxiliary qubits (control qubits) is $|000000000\rangle$ which is a string of nine zeros, each corresponding to one crosspoint in the crossbar. Recall that control qubits are set to 1 for the cross state and to 0 for the through state. The output state of the auxiliary qubits is indicated beside the crosspoints. Thus, the output state of the auxiliary qubits is $|000110111\rangle$ where the bits are written in order from top to bottom and left to right, e.g., the third crosspoint for input 4 is set to a through state and this is the sixth crosspoint in traversal order, so the sixth bit in the output state is 0.

Output for Capacity Exceeding Input Patterns

So far we have shown that in a fat-slim or banded QSC(n, m), self-routing can be used to concentrate any capacity achieving input assignment pattern. We now present the case when the input pattern exceeds the capacity of the crossbar.

For a self-routing QSC(n, m), consider a capacity exceeding input concentration pattern with r, $(r > m)$ valid packets. The ordered set of inputs with packets to concentrate is $\mathcal{X} = \mathcal{X}_1^m \cup \mathcal{X}_{m+1}^r$ where $\mathcal{X}_1^m = \{x_1, \ldots, x_m\}$ and $\mathcal{X}_{m+1}^r = \{x_{m+1}, \ldots, x_r\}$. Since inputs are routed in increasing

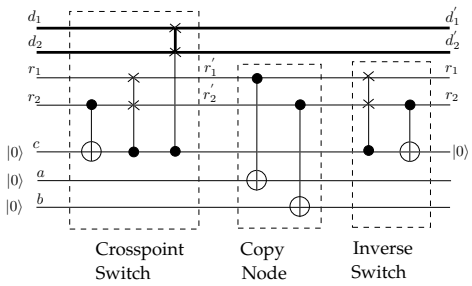


Fig. 12: Circuit for restoring the control quantum bit c.

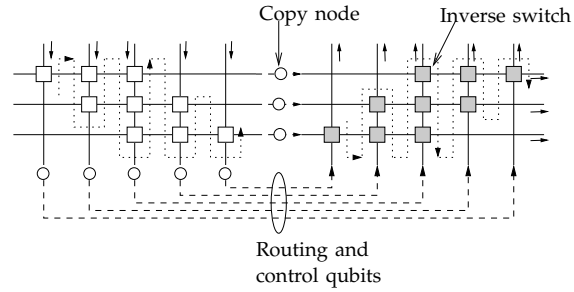


Fig. 13: A Banded QSC(5,3) with additions for restoring the control quantum bits.

order, all inputs in $\mathcal{X}_1^m$ are concentrated to the m outputs, i.e., $\mathcal{Y} = \{y_1, \ldots, y_m\} = \mathcal{O}$. For input x_{m+1}: $y_{m+1} = \min_z\{z \in \mathcal{A}_{x_{m+1}} \cap (\mathcal{Y})'\} = \min_z\{z \in \mathcal{A}_{x_{m+1}} \cap \emptyset\} = \emptyset$. Similarly for the other inputs in $\mathcal{X}_{m+1}^r$, the matching output is $\emptyset$, i.e., $\{y_{m+1}, \ldots, y_r\} = \emptyset$. If $y_i = \emptyset$, then all crosspoints in the column for the corresponding input x_i are set to cross state and the packet comes out on the bottom, which is at output $x_i + m$. Hence, the m lowest numbered inputs are concentrated and the rest are connected to corresponding output at the bottom.

8 Restoring Auxiliary Control Quantum Bits

Quantum information can be encoded in many different ways, such as the spin component of basic particles like electrons or protons, or in the polarization of photons. But, such particles can interact with the environment which leads to a corruption of their quantum state, a process known as *decoherence*. Decoherence can be viewed as a measurement of a superposed quantum state which collapses it to one of its basis states. This leads to a loss of information, but for a quantum circuit, this information loss can be overcome if the ancillary qubits used as control qubits are restored back to their original states, so that a corruption of their state does not affect the observed quantum data. We now give a method to restore the state of the auxiliary bits back to their original state, i.e., $|0\rangle$. For a single quantum crosspoint we can restore the control quantum bit back to the state $|0\rangle$ as shown in Figure 12. The mapping performed is:

$$|(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |00\rangle_{ab} \xrightarrow[\textit{Through}]{\textit{Crosspoint}} |(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |00\rangle_{ab} \tag{26}$$

$$\xrightarrow{\textit{Copy}} |(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |r_1 0\rangle_{ab} \tag{27}$$

$$\xrightarrow{\textit{Inverse}} |(r_1, d_1), (0, d_2)\rangle\, |0\rangle_c\, |r_1 0\rangle_{ab} \tag{28}$$

$$|(r_1, d_1), (1, d_2)\rangle\, |0\rangle_c\, |00\rangle_{ab} \xrightarrow[\textit{Cross}]{\textit{Crosspoint}} |(1, d_2), (r_1, d_1)\rangle\, |1\rangle_c\, |00\rangle_{ab} \tag{29}$$

$$\xrightarrow{\textit{Copy}} |(1, d_2), (r_1, d_1)\rangle\, |1\rangle_c\, |1\, r_1\rangle_{ab} \tag{30}$$

$$\xrightarrow{\textit{Inverse}} |(r_1, d_2), (1, d_1)\rangle\, |0\rangle_c\, |1\, r_1\rangle_{ab} \tag{31}$$

At the output of the quantum crosspoint the two CNOT gates in the copy circuit copy the values of bits r_1 and r_2 onto a and b respectively. This can be seen in Eqn. (27) and Eqn. (30). The inverse switch then does a controlled swap of the two routing bits r_1 and r_2 before restoring c back to its original state as can be seen in Eqn. (28) and Eqn. (31). Note that when the bit c is 0 at the output of the quantum crosspoint switch then the restoring portion does not modify anything (see Eqns. (26)-(28)) as the corresponding auxiliary qubit is already in the state $|0\rangle$.

On measurement at the output we determine valid (not valid) packets by observing their associated routing bit as 1 (0). But note that on final measurement in Figure 12 the routing bits may not correspond to the data part of their packets, this is seen in Eqn. (31) where r_1, d_2 and $1, d_1$ are together instead of being $1, d_2$ and r_1, d_1. But the copying operation ensures that we have a copy of the correct routing bits and can use these to distinguish the valid packets, for example, in Eqn. (31) the correct values of the routing bits at the output are 1 for the upper packet and r_1 for the lower packet and these are present in the correct order, $1, r_1$, on qubits a and b. Thus we can now consider a as the routing qubit for the packet at the upper output and b as the routing qubit for the packet at the lower output.

This circuit restores the control qubit for a single crosspoint. For the entire self-routing QSC we need a mirror image of the sparse crossbar concentrator concatenated with the QSC after the copying of the routing bits is done at the output to restore the control qubits. This is shown in Figure 13. Only the routing qubits are involved in restoring the state of the control qubits, hence only these qubits are forwarded to the next stage after the QSC and are shown by dashed lines. The dotted lines show the order of traversal of crosspoints and inverse switches.

9 Cost Analysis

We have introduced quantum concentrators and presented two designs of such concentrators using self-routing sparse crossbars. The complexity of these quantum concentrators can be computed as follows:

We need, per crosspoint, one multi-qubit switch gate for swapping the $n_d + 1$ bit packets and one CNOT gate for setting the auxiliary control bit. A switch gate for swapping one quantum bit packets can be implemented using two CNOT gates and one CCNOT gate. Hence, we need $2(n_d + 1)$ CNOT and $n_d + 1$ CCNOT gates for the multi-qubit switch gate for a total of $2n_d + 3$ CNOT and $n_d + 1$ CCNOT gates per crosspoint. Therefore, each quantum concentrator design we presented uses $m(n - m + 1)(2n_d + 3)$ CNOT gates, $m(n - m + 1)(n_d + 1)$ CCNOT gates and $m(n - m + 1)$ auxiliary quantum bits.

For the restoring stage, there are $m(n-m+1)$ inverse switches and $n+m$ copy nodes. Each inverse switch has one switch gate for single qubits and one CNOT gate, which sums up to three CNOT and one CCNOT gate. Each copy node has two CNOT gates and two extra qubits. Thus the total cost for restoring the control qubits is $3m(n-m+1)+2(n+m)$ CNOT gates, $m(n-m+1)$ CCNOT gates and $2(m+n)$ extra qubits. Therefore, the overall cost for a QSC(n,m) is $m(n-m+1)(2n_d+6)+2(n+m)$ CNOT gates, $m(n-m+1)(n_d+2)$ CCNOT gates and $m(n-m+1)+2(n+m)$ auxiliary quantum bits.

The depth of a QSC(n,m) is given by the maximum possible number of crosspoints between an input and an output. It is easy to see that the longest input-output path is between input 1 and output m. For the fat-slim QSC(n,m), this path length is $(n-m+1)+m-1=n$ crosspoints and for the banded QSC(n,m) the path length is $(n-m+1)+(m-1)+(m-1)=n+m-1$ crosspoints. Hence the depth of fat-slim QSC(n,m) is n and the depth of banded QSC(n,m) is $n+m-1$. The time required to self-route is upper bounded by the depth of the concentrator, thus, self-routing on a fat-slim QSC(n,m) has $O(n)$ delay and self-routing on a banded QSC(n,m) has $O(n+m)$ delay.

10 Concluding Remarks

Our results demonstrate that quantum principles can be applied to concentration problems in packet switching. In proving that fat-slim and banded crossbar concentrators are self-routable, we have assumed input quantum assignments consisting of single patterns of classical packets. This proves that classical fat-slim and banded crossbar concentrators are self-routable as well. We also note that in our self-routing algorithm, when the capacity of m packets is exceeded, only the m lowest numbered inputs have their packets concentrated. This introduces the issue of fairness in routing. If the capacity is exceeded, one way to ensure all inputs have an equal probability for concentration is to create an equal superposition of the input packets at the outputs of a crosspoint whenever both incoming packets have their routing bits equal to 1. The resulting output quantum concentration assignment would then contain superposed concentration patterns in which valid packets from inputs other than the lowest m are present.

Another direction to be explored further is the tradeoff between delay and fanout. We see that the delay varies with topology, as the more "balanced" banded crossbar having a larger delay of $O(n+m)$ than the $O(n)$ delay for the fat-slim crossbar. This seems to be a consequence of the sequential nature of the routing algorithm. The dependence of delay on routing strategy and topology is thus another direction for further research.

We have established the self-routability of two families of sparse crossbar concentrators, namely the fat-slim and banded crossbars. Finding other topologies which allow self-routing remains an open question. The density of (n, m)-sparse crossbar concentrators among all $n \times m$ crossbars is known [30] as are equivalence relations between different classical sparse crossbar concentrator topologies [25]. A similar approach could be employed to characterize self-routable crossbar concentrators and find structures which belong to this class. In our algorithm, inputs were routed in increasing order of their indicies. By changing the order in which this routing is done, we could find other structures which allow self-routing. A trivial example would be to route in decreasing order of inputs on a fat-slim crossbar with inputs 1 to $(n-m)$ comprising the fat section and inputs $(n-m+1)$ to n the slim section. In particular, an interesting direction for further investigation would be to determine if there exist self-routing regular sparse crossbar concentrators, i.e., those with fixed out-degree inputs and in-degree outputs. All these questions apply to both classical and quantum sparse crossbar concentrators.

Acknowledgements. We thank the referees for their significant suggestions that helped improved the presentation of the paper.

Appendix A

Proof for Theorem 3

Proof: In the ordered r-input set $\mathcal{X}$, let the first k ($k \leq r$) inputs belong to the slim section and the rest to the fat section.

If $x_1 \in \mathcal{I}_S$, we get

$$\begin{aligned} y_1 &= \min_z \{z \in \mathcal{A}_{x_1} \cap \mathcal{Y}_0'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \mathcal{A}_{x_1} \cap \mathcal{O}\} && \text{(as } \mathcal{Y}_0 = \emptyset) \\ &= \min_z \{z \in \mathcal{A}_{x_1}\} = \pi(x_1) && \text{(from Eqn. (20a))} \end{aligned} \tag{32}$$

Hence, $y_1 \neq \emptyset$ and $\mathcal{Y}_1 = \{\pi(x_1)\}$ is the set of matched outputs after routing on the first input.

Similarly for $x_2 \in \mathcal{I}_S$ we get

$$\begin{aligned} y_2 &= \min_z \{z \in \mathcal{A}_{x_2} \cap (\mathcal{Y}_1)'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \{\pi(x_2)\} \cap \{\pi(x_1)\}'\} && \text{(from Eqn. (20a))} \\ &= \min_z \{z \in \{\pi(x_2)\}\} = \pi(x_2) && \text{(as } \pi(x_1) \neq \pi(x_2)) \end{aligned}$$

Continuing this way we get $\mathcal{Y}_1^k = \{\pi(x_1), \pi(x_2), \ldots, \pi(x_k)\}$. (33)

Now since $\mathcal{X}$ is an ordered set of distinct inputs, all the elements of $\mathcal{Y}_1^k$ are distinct and hence form a matching for the k slim inputs. If $k = r$ then the proof is complete, else for $x_{k+1} \in \mathcal{I}_F$ we get

$$\begin{aligned} y_{k+1} &= \min_z\{z \in \mathcal{A}_{x_{k+1}} \cap (\mathcal{Y}_1^k)'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \mathcal{O} \cap \{\pi(x_1), \ldots, \pi(x_k)\}'\} && \text{(from Eqns. (20b) and (33))} \\ &= \min_z\{z \in \{\pi(x_1), \ldots, \pi(x_k)\}'\} && (34) \end{aligned}$$

Note $|\{\pi(x_1), \ldots, \pi(x_k)\}'| = m - k$. Let $\mathcal{B} = \{b_1, \ldots, b_{m-k}\}$ where $b_i \in (\mathcal{Y}_1^k)',\ i = 1, \ldots, m - k,$ such that $b_1 < b_2 < \cdots < b_{m-k}$. Thus, $\mathcal{B}$ is an ordered version of $(\mathcal{Y}_1^k)' = \{\pi(x_1), \ldots, \pi(x_k)\}'$ with elements arranged in increasing order of magnitude, $|\mathcal{B}| = m - k$. From Eqn. (34), $y_{k+1} = \min_z\{z \in \mathcal{B}\} = b_1$.

Also, $k \leq r \leq m$ and if $k = r = m$, i.e., all the slim inputs are concentrated then $y_{k+1} = \emptyset$ and $\mathcal{Y}_1^k$ is the matching corresponding to the concentration. If $k < r$ then clearly $y_{k+1} = b_1 \neq \emptyset$ as then $\mathcal{B} \neq \emptyset$. Thus $\mathcal{Y}_1^{k+1} = \mathcal{Y}_1^k \cup \{b_1\} = \{\pi(x_1), \ldots, \pi(x_k), b_1\}$.

For input $y_{k+2} \in \mathcal{I}_F$ we get

$$\begin{aligned} y_{k+2} &= \min_z\{z \in \mathcal{A}_{x_{k+2}} \cap (\mathcal{Y}_1^{k+1})'\} \\ &= \min_z\{z \in \mathcal{O} \cap (\mathcal{Y}_1^k \cup \{b_1\})'\} \\ &= \min_z\{z \in (\mathcal{Y}_1^k)' \cap \{b_1\}'\} \\ &= \min_z\{z \in \mathcal{B} \cap \{b_1\}'\} = b_2 \end{aligned}$$

Continuing in the same fashion we get $\mathcal{Y}_{k+1}^r = \{b_1, \ldots, b_{r-k}\}$. Now, $|\mathcal{B}| = m - k$, i.e., $k = m - |\mathcal{B}|$. Thus, $y_{k+i} = b_i = b_{(k+i)-k} = b_{k+i-(m-|\mathcal{B}|)}$, $i = 1, \ldots, r - k$. Hence, $y_i = b_{i-(m-a)},\ i = k + 1, \ldots, r$, $a = |\mathcal{B}|$. Therefore, by Theorem 2 the fat-slim QSC(n, m) is self routing. □

Appendix B

Proof for Theorem 4

Proof: In $\mathcal{X}$ let k_1 inputs belong to $\mathcal{I}_U$, k_2 inputs belong to $\mathcal{I}_T$ and the rest $r-(k_1+k_2)$ inputs belong to $\mathcal{I}_L$, i.e., $\{x_1, \ldots, x_{k_1}\} \subseteq \mathcal{I}_U$, $\{x_{k_1+1}, \ldots, x_{k_1+k_2}\} \subseteq \mathcal{I}_T$ and $\{x_{k_1+k_2+1}, \ldots, x_r\} \subseteq \mathcal{I}_L$.

Case I: $n \geq 2m - 1$

For input $x_1 \in \mathcal{I}_U$:

$$\begin{aligned} y_1 &= \min_z\{z \in \mathcal{A}_{x_1} \cap \mathcal{Y}_0'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \mathcal{A}_{x_1} \cap \mathcal{O}\} && \text{(as } \mathcal{Y}_0 = \emptyset) \\ &= \min_z\{z \in \{1, 2, \ldots, x_1\}\} = 1 && \text{(from Eqn. (21a))} \end{aligned}$$

Similarly for $x_2 \in \mathcal{I}_U$

$$\begin{aligned} y_2 &= \min_z\{z \in \mathcal{A}_{x_2} \cap (\mathcal{Y}_1)'\} \\ &= \min_z\{z \in \{1,\ldots,x_2\} \cap \{1\}'\} \\ &= \min_z\{z \in \{2,\ldots,x_2\}\} = 2 \end{aligned}$$

Proceeding this way for all inputs in $\mathcal{I}_U$ we get $\mathcal{Y}_1^{k_1} = \{1,2,\ldots,k_1\}$.

For input $x_{k_1+1} \in \mathcal{I}_T$:

$$\begin{aligned} y_{k_1+1} &= \min_z\{z \in \mathcal{A}_{x_{k_1+1}} \cap (\mathcal{Y}_1^{k_1})'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \{1,\ldots,m\} \cap \{1,\ldots,k_1\}'\} && \text{(from Eqn. (21b))} \\ &= \min_z\{z \in \{k_1+1,\ldots,m\}\} = k_1+1 \end{aligned}$$

We can get similar results for all other inputs in $\mathcal{I}_T$. Thus, $\mathcal{Y}_1^{k_1+k_2} = \{1,\ldots,k_1+k_2\}$.

Also when $1 \le i \le k_1+k_2$, $x_i \in \mathcal{I}_U \cup \mathcal{I}_T$

$$\begin{aligned} &\Rightarrow && x_i \le n-m+1 && \text{(from Eqns. (21a) and (21b))} \\ &\Rightarrow && x_i - n + m \le 1 \\ &\Rightarrow && x_i - n + m \le i = y_i \end{aligned}$$

Hence, $y_i = i = \max(i,\ x_i - n + m)$, for all $1 \le i \le k_1+k_2$.

For input $x_{k_1+k_2+1} \in \mathcal{I}_L$:

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z\{z \in \mathcal{A}_{x_{k_1+k_2+1}} \cap (\mathcal{Y}_1^{k_1+k_2})'\} \\ &= \min_z\{z \in \{x_{k_1+k_2+1}-n+m,\ldots,m\} \cap \{k_1+k_2+1,\ldots,m\}\} && (35) \\ &= \min_z\{z \in \{\max(x_{k_1+k_2+1}-n+m,\ k_1+k_2+1),\ldots,m\}\} && (36) \\ &= \max(k_1+k_2+1,\ x_{k_1+k_2+1}-n+m) && (37) \end{aligned}$$

where Eqn. (35) follows from Eqn. (21c) and Eqn. (36) follows from the fact that we are taking the intersection of 2 sets both of which cover continuous intervals of outputs up to output m. We now use induction to prove the rest of the theorem.

Induction assumption:

$$y_i = \max(i,\ x_i - n + m), \quad \text{for all } i = k_1+k_2+1,\ldots,j-1 \text{ where } j \le r. \tag{38}$$

Need to prove: $y_j = \max(j,\ x_j - n + m)$

Proof for Induction: We have already proved the base case for $i = k_1 + k_2 + 1$. We will first show that, for y_i's chosen according to Eqn. (38), $y_{i-1} < y_i$.

Note $y_{i-1} = \max(i-1, x_{i-1} - n + m)$ and $y_i = \max(i, x_i - n + m)$, thus we get the following cases:

1) $y_{i-1} = i - 1$: We get the followings series of inequalities:

$$\begin{aligned} \max(i, x_i - n + m) &\geq i \\ \Rightarrow y_i &\geq i \quad (\text{as } y_i = \max(i, x_i - n + m)) \\ &> i - 1 = y_{i-1} \end{aligned}$$

2) $y_{i-1} = x_{i-1} - n + m$: We get the followings series of inequalities:

$$\begin{aligned} \max(x_i - n + m, i) &\geq x_i - n + m \\ \Rightarrow y_i &\geq x_i - n + m \quad (\text{as } y_i = \max(i, x_i - n + m)) \\ &> x_{i-1} - n + m = y_{i-1} \quad (\text{as } x_i > x_{i-1}) \end{aligned}$$

Thus $y_i > y_{i-1},\ i = k_1 + k_2 + 2, \ldots, j - 1$.

We know that $\mathcal{Y}_1^{k_1+k_2} = \{1, \ldots, k_1 + k_2\}$ is monotonically increasing, $\mathcal{Y}_{k_1+k_2+1}^{j-1}$ is monotonically increasing and $y_{k_1+k_2+1} = \max(k_1 + k_2 + 1, x_{k_1+k_2+1} - n + m)) > k_1 + k_2 = y_{k_1+k_2}$.

Thus $\mathcal{Y}_1^{j-1} = \mathcal{Y}_1^{k_1+k_2} \cup \mathcal{Y}_{k_1+k_2+1}^{j-1}$ is monotonically increasing, i.e., $y_1 < y_2 < \cdots < y_{j-1}$.

By Lemma 2, $y_j = \min_z \{z \in \mathcal{A}_{x_j} \cap (\mathcal{Y}_1^{j-1})'\}$ (39)

By induction assumption $y_{j-1} = \max(j-1, x_{j-1} - n + m)$.

Case 1: $y_{j-1} = x_{j-1} - n + m$

By monotonicity of $\mathcal{Y}_1^{j-1}$ we get $\max_z(z \in \mathcal{Y}_1^{j-1}) = y_{j-1} = x_{j-1} - n + m$. Thus

$$(\mathcal{Y}_1^{j-1})' = \mathcal{Z} \cup \{x_{j-1} - n + m + 1, \ldots, m\}$$

$$\text{where } \mathcal{Z} \subseteq \{k_1 + k_2 + 1, \ldots, x_{j-1} - n + m - 1\}$$

Therefore,

$$\begin{aligned} \mathcal{A}_{x_j} \cap (\mathcal{Y}_1^{j-1})' &= \{x_j - n + m, \ldots, m\} \cap [\mathcal{Z} \cup \{x_{j-1} - n + m + 1, \ldots, m\}] \\ &= \emptyset \cup \{x_j - n + m, \ldots, m\} \cap \{x_{j-1} - n + m + 1, \ldots, m\} \\ &= \{(\max(x_j - n + m,\ x_{j-1} - n + m + 1)), \ldots, m\} \\ &= \{x_j - n + m, \ldots, m\} \quad (\text{as } x_j \geq x_{j-1} + 1) \end{aligned} \tag{40}$$

Substituting in Eqn. (39) we get

$$y_j = \min_z\{z \in \{x_j - n + m, \ldots, m\}\} = x_j - n + m \tag{41}$$

Now $y_{j-1} = \max(x_{j-1} - n + m,\ j-1) = x_{j-1} - n + m$. Thus

$$\begin{aligned} x_{j-1} - n + m &\geq j - 1 \\ \Rightarrow x_{j-1} - n + m + 1 &\geq j \\ \Rightarrow x_j - n + m &\geq j \quad (\text{as } x_j \geq x_{j-1} + 1) \end{aligned} \tag{42}$$

From Eqns. (41) and (42)

$$y_j = \max(x_j - n + m,\ j) \tag{43}$$

Case 2: $y_{j-1} = j - 1$

$\mathcal{Y}_1^{j-1}$ is monotonically increasing, i.e., $y_1 < y_2 < \cdots < y_{j-1}$ and $y_{j-1} = j - 1$

$$\begin{aligned} \Rightarrow \mathcal{Y}_1^{j-1} &= \{1, 2, \ldots, j-1\} \\ \Rightarrow (\mathcal{Y}_1^{j-1})' &= \{j, \ldots, m\} \end{aligned}$$

Substituting in Eqn. (39) we get

$$\begin{aligned} y_j &= \min_z\{\{x_j - n + m, \ldots, m\} \cap \{j, \ldots, m\}\} \\ &= \min_z\{\max(x_j - n + m,\ j), \ldots, m\} \\ &= \max(x_j - n + m,\ j) \end{aligned} \tag{44}$$

From Eqn. (43) and Eqn. (44) $y_j = \max(x_j - n + m,\ j)$ and proof for the induction is complete.

Case II: $m \leq n < 2m - 1$

k_1 inputs in S belong to $\mathcal{I}_U$, i.e., $\{x_1, \ldots, x_{k_1}\} \subseteq \mathcal{I}_U$. For $y_1, \ldots, y_{k_1}$ the proof is exactly the same as for the case $n \geq 2m-1$ and we get $\mathcal{Y}_1^{k_1} = \{1, 2, \ldots, k_1\}$ and hence, $y_i = i = \max(i,\ x_i - n + m)$, $i = 1, \ldots, k_1$.

The next k_2 inputs in $\mathcal{X}$ belong to $\mathcal{I}_T$, i.e., $\{x_{k_1+1}, \ldots, x_{k_1+k_2}\} \subseteq \mathcal{I}_T$.

For y_{k_1+1} we get

$$\begin{aligned} y_{k_1+1} &= \min_z\{z \in \mathcal{A}_{x_{k_1+1}} \cap (\mathcal{Y}_1^{k_1})'\} && \text{(from Lemma 2)} \\ &= \min_z\{z \in \{x_{k_1+1} - n + m, \ldots, x_{k_1+1}\} \cap \{1, \ldots, k_1\}'\} && \text{(from Eqn. (22b))} \\ &= \min_z\{z \in \{x_{k_1+1} - n + m, \ldots, x_{k_1+1}\} \cap \{k_1 + 1, \ldots, m\}\} \end{aligned}$$

Both sets in the intersection contain a continuous series of outputs. Now obviously, $x_{k_1+1} \leq m$. Also, $k_1 + 1 \leq x_{k_1+1}$ as

$$\begin{aligned} k_1 &\leq n - m + 1 && \text{(from Eqn. (22a))} \\ \Rightarrow k_1 + 1 &\leq n - m + 2 \\ &\leq x_{k_1+1} && \text{(from Eqn. (22b))} \end{aligned}$$

Thus,
$$\begin{aligned} y_{k_1+1} &= \min_z \{z \in \{\max(k_1 + 1, x_{k_1+1} - n + m), \ldots, x_{k_1+1}\}\} \\ &= \max(k_1 + 1, x_{k_1+1} - n + m) \end{aligned}$$

Treating this as the base case, we can use an induction argument similar to that employed for inputs in $\mathcal{I}_L$ for the case $n \geq 2m - 1$ to show that

$$\begin{aligned} y_i &= \max(i,\ x_i - n + m), \quad k_1 + 1 \leq i \leq k_1 + k_2 \\ \Rightarrow y_i &= \max(i,\ x_i - n + m), \quad x_i \in \mathcal{I}_T \end{aligned} \tag{45}$$

For input $x_{k_1+k_2+1}$ we get

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z \{z \in \mathcal{A}_{x_{k_1+k_2+1}} \cap (\mathcal{Y}_1^{k_1+k_2})'\} && \text{(from Lemma 2)} \\ &= \min_z \{z \in \{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap (\mathcal{Y}_1^{k_1+k_2})'\} \end{aligned} \tag{46}$$

As $y_{k_1+k_2} = \max(k_1 + k_2,\ x_{k_1+k_2} - n + m)$, we get the following two cases:

Case 1: $y_{k_1+k_2} = k_1 + k_2$. Since $y_1 < y_2 < \cdots < y_{k_1+k_2}$, and $y_{k_1+k_2} = k_1 + k_2$

$$\begin{aligned} \mathcal{Y}_1^{k_1+k_2} &= \{1, 2, \ldots, k_1 + k_2\} \\ \Rightarrow (\mathcal{Y}_1^{k_1+k_2})' &= \{k_1 + k_2 + 1, \ldots, m\} \end{aligned} \tag{47}$$

Substituting Eqn. (47) in Eqn. (46) we get

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z \{z \in \{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap \{k_1 + k_2 + 1, \ldots, m\}\} \\ &= \min_z \{z \in \{\max(x_{k_1+k_2+1} - n + m,\ k_1 + k_2 + 1), \ldots, m\}\} \\ &= \max(x_{k_1+k_2+1} - n + m,\ k_1 + k_2 + 1). \end{aligned} \tag{48}$$

Case 2: $y_{k_1+k_2} = x_{k_1+k_2} - n + m$. Since $y_1 < \cdots < y_{k_1+k_2}$, $\max_z(z \in \mathcal{Y}_1^{k_1+k_2}) = y_{k_1+k_2} = x_{k_1+k_2} - n + m$.

$\Rightarrow (\mathcal{Y}_1^{k_1+k_2})' = \mathcal{Z} \cup \{x_{k_1+k_2} - n + m + 1, \ldots, m\},\ \mathcal{Z} \subseteq \{k_1 + k_2 + 1, \ldots, x_{k_1+k_2} - n + m - 1\}$.

Substituting this in Eqn. (48) we get

$$\begin{aligned} y_{k_1+k_2+1} &= \min_z\{z \in \{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap (\mathcal{Z} \cup \{x_{k_1+k_2} - n + m + 1, \ldots, m\})\} \\ &= \min_z\{z \in \emptyset \cup (\{x_{k_1+k_2+1} - n + m, \ldots, m\} \cap \{x_{k_1+k_2} - n + m + 1, \ldots, m\})\} \\ &= \min_z\{z \in \{\max(x_{k_1+k_2} - n + m + 1,\ x_{k_1+k_2+1} - n + m), \ldots, m\}\} \\ &= \max(x_{k_1+k_2} - n + m + 1,\ x_{k_1+k_2+1} - n + m) \\ &= x_{k_1+k_2+1} - n + m \qquad (\text{as } x_{k_1+k_2+1} \geq x_{k_1+k_2} + 1) \end{aligned} \tag{49}$$

Also,

$$\begin{aligned} x_{k_1+k_2+1} - n + m &\geq x_{k_1+k_2} + 1 - n + m \\ &= y_{k_1+k_2} + 1 \\ &\geq k_1 + k_2 + 1 \end{aligned} \tag{50}$$

where Eqn. (50) follows from the fact that $y_{k_1+k_2} = \max(x_{k_1+k_2} - n + m,\ k_1 + k_2)$. From Eqns. (49) and (50), $y_{k_1+k_2+1} = \max(k_1 + k_2 + 1,\ x_{k_1+k_2+1} - n + m)$.

We can now use an induction argument for rest of the inputs in $\mathcal{I}_L$ similar to the case for $n \geq 2m - 1$ to show that

$$y_i = \max(i,\ x_i - n + m), \quad k_1 + k_2 + 1 \leq i \leq r$$

$$\Rightarrow y_i = \max(i,\ x_i - n + m), \quad x_i \in \mathcal{I}_L.$$

Thus, by Theorem 2 the banded QSC(n, m) is self-routing. □

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Computing* , vol. 26, pp. 1484–1509, Oct. 1997.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 212–219, May 1996.

[3] J. Preskill. (1998) Physics 229: Advanced mathematical methods of physics – quantum computation and information. [Online]. Available: http://www.theory.caltech.edu/people/preskill/ph229

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, Sept. 2000.

[5] D. Vion *et al.*, "Manipulating the quantum state of an electrical circuit," *Science*, vol. 296, p. 886, May 2002.

[6] B. Kane, "A silicon-based nuclear spin quantum computer," *Nature*, vol. 393, pp. 133-137, May 1998.

[7] D. Copsey *et al.*, "Toward a scalable, silicon-based quantum computing architecture," *IEEE Journal Of Selected Topics in Quantum Electronics*, vol. 9, no. 6, pp. 1552–1569, Nov-Dec 2003.

[8] M. Oskin *et al.*, "Building quantum wires: the long and the short of it," in *Proceedings of 30th Annual International Symposium on Computer Architecture* , pp. 374–385, June 2003.

[9] W. Wootters and W. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802–803, Oct. 1982.

[10] N. Isailovic *et al.*, "Datapath and control for quantum wires," *ACM Transactions on Architecture and Code Optimization*, vol. 1, no. 1, pp. 34–61, March 2004.

[11] T. S. Metodi, D. D. Thaker, and A. W. Cross, "A quantum logic array microarchitecture: Scalable quantum data movement and computation," in *Proceedings of 38th Annual IEEE/ACM International Symposium on Microarchitecture. MICRO-38*, pp. 305–318, Nov. 2005.

[12] C. Moore and M. Nilsson, "Parallel quantum computation and quantum codes," *SIAM Journal on Computing*, vol. 31, no. 3, pp. 799–815, 2001.

[13] I. M. Tsai and S.-Y. Kuo, "Digital switching in the quantum domain," *IEEE Transactions on Nanotechnology*, vol. 1, no. 3, pp. 154–164, Sept. 2002.

[14] M. K. Shukla, R. Ratan, and A. Y. Oruç, "A quantum self-routing packet switch," in *Proceedings of the 38th Annual Conference on Information Sciences and Systems. CISS'04*, Princeton, NJ, pp. 484–489, Mar. 2004.

[15] ——, "The quantum baseline network," in *Proceedings of the 39th Annual Conference on Information Sciences and Systems. CISS'05*, Baltimore, MD, Mar. 2005.

[16] S. T. Cheng and C. Y. Wang, "Quantum switching and quantum merge sorting," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 2, pp. 316–325, Feb. 2006.

[17] C.-C. Sue, W.-R. Chen, and C.-Y. Huang, "Design and analysis of a fully non-blocking quantum switch," in *First International Conference on Innovative Computing, Information and Control. ICICIC '06*, vol. 2, 2006, pp. 421–424.

[18] C.-C. Sue, "An enhanced universal n x n fully nonblocking quantum switch," *IEEE Transactions on Computers*, vol. 58, no. 2, pp. 238–250, 2009.

[19] R. Ratan and A. Y. Oruç, "Quantum switching networks with classical routing," in *Proceedings of the 41st Annual Conference on Information Sciences and Systems. CISS'07*, Baltimore, MD, Mar. 2007.

[20] S.-T. Cheng, C.-Y. Wang, and M.-H. Tao, "Quantum communication for wireless wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 7, July 2005.

[21] L. A. Bassalygo and M. S. Pinsker, "Complexity of an optimum nonblocking switching network without reconnections," *Problems of Information Transmission*, vol. 9, no. 1, 1974.

[22] L. A. Bassalygo, "Asymptotically optimal switching circuits," *Problems of Information Transmission*, vol. 17, no. 3, pp. 206–211, 1981.

[23] C. E. Shannon, "Memory requirements in a telephone exchange," *Bell System Technical Journal*, vol. 29, pp. 343–349, 1950.

[24] A. Yavuz Oruç and H. M. Huang, "Crosspoint complexity of sparse crossbar concentrators," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1466–1471, Sept. 1996.

[25] W. Guo and A. Y. Oruç, "Regular sparse crossbar concentrators," *IEEE Transactions on Computers*, vol. 47, no. 3, pp. 363–368, Mar. 1998.

[26] W. Guo, "Design and optimization of switching fabrics for ATM networks and parallel computer systems," Ph.D. dissertation, Dept. of Electrical Engg., University of Maryland, College Park, MD, Aug. 1996.

[27] D. Deutsch, "Quantum computational networks," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 425, no. 1868, pp. 73–90, Sept. 1989.

[28] S. Nakamura and G. M. Masson, "Lower bounds on crosspoints in concentrators," *IEEE Transactions on Computers*, vol. C-31, no. 12, pp. 1173–1179, Dec. 1982.

[29] M. K. Shukla and A. Y. Oruç, "Multicasting in quantum switching networks," *IEEE Transactions on Computers*, To appear.

[30] E. Günduzhan and A. Y. Oruç, "Structure and density of sparse crossbar concentrators," in *DIMACS Series in Discrete Mathematics and Computer Science*. Advances in Switching Networks, vol. 42, pp. 169–180, Oct. 1998.

Changes Made To Address The Reviewers' Comments

We thank both referees for their significant suggestions that helped improved the presentation of our paper.

Reviewer 1:

Comment 1. The paper considers the quantum circuit design and routing of an n x m network, called a quantum concentrator that can direct quantum bit packets in arbitrary quantum states from any of its k inputs to some of its k outputs.

Response: We thank the reviewer for accurately characterizing our contributions.

Comment 2. The authors address interesting issues about quantum operation like reversibility and localized self-routing and give a rigorous proof that quantum fat-slim and banded sparse crossbar concentrators are self-routable.

Response: We thank the reviewer for accurately characterizing our contributions.

Comment 3. The self-routing algorithms described in the paper can be used for both quantum and classical sparse crossbar concentrators by the linearity property of all quantum systems.

Response: We thank the reviewer for accurately characterizing our contributions.

Comment 4. A more thorough survey in the literature is suggestion. To my knowledge, the work about quantum routing in wireless domain has been published.

It seems to me that the linkage of the proposed paper to the wireless domain might be possible in the future. Authors are suggested to include the following paper to the related work and background.

"Quantum Communication for Wireless WAN Networks," IEEE Journal on Selected Areas in Communications, 2005.

Response: We thank the reviewer for making our literature survey more extensive. We have added four more citations, including the one that was suggested by the referee. The new citations are listed below (References 20, 17, 18,12]:

S-T. Cheng, C-Y. Wang, and M-H. Tao. "Quantum Communication for Wireless WAN Networks, " IEEE Journal of Selected Areas in Communications, pp. 1424-1432, July 2005.

C-C. Sue. "An enhanced universal nxn fully nonblocking quantum switch," IEEE Transactions on Computers, pp. 238-250, February 2009.

C-C. Sue, W-R. Chen, and C-Y. Huang. "Design and analysis of a fully nonblocking quantum switch," First International Conference on Innovative Computing, Information and Control, Vol. 2, pp. 421–424, 2006.

C. Moore and M. Nilsson, "Parallel quantum computation and quantum codes," SIAM Journal on Computing, vol. 31, no. 3, pp. 799–815, 2001.

Comment 5. Some typos in the paper:

Page 11, line 33: 1 <= i <= n shall be changed to 1 <= i <= r.

Notations shall be consistent: For example, at page 30, If X^r_m+1 represents { x_m+1, x_m+2, .. x_r}, then {x_1, x_2, ... x_m} shall be represented by X^m_1 instead of X_m.

Similarly, at page 18, if Y^b_a is {y_a, y_a+1, .., y_b}, then please {y_1, y_2, ..., y_b} shall be represented by Y^b_1 instead of Y_b.

Authors are required to check through the paper and make the notations consistent.

Response: We thank the reviewer for pointing out the typographical errors. We fixed the ones he suggested and also went over the notation more thoroughly to fix other typographical errors as well.

Reviewer 2:

Comment 1. The authors focus on the quantum concentrator switches that realize unordered connections between a set of inputs and a set of outputs. However, in general, quantum switching networks like the research [10-13] and a related article in Feb. 2009, Trans. On Computer, realize ordered connection maps, i.e., the destination output for a particular input is specified. The difficulty faced by the non-blocking routing in these literatures is not seen in this paper. The authors should provide the pros and cons for these two different types of connection requirements.

Response: We thank the reviewer for suggesting a comparison of ordered v.s. unordered connection networks. We added comments in the introduction section (bottom of page 2 and beginning of page 3) to address this issue where we indicate the relation between ordered and unordered connection networks and mention their limitations and pros/cons.

Comment 2. On page 17, the authors said "…hence we need to introduce appropriate delay to make them all equal…." However, throughout the whole paper, no explicit mechanisms are provided to solve this problem. Different (n,m) concentrator structures should be deployed with different delays. How to guarantee the delay is appropriate for multi-stage quantum circuits is not trivial. The authors should elaborate on this problem.

Response: We thank the reviewer for pointing out this issue. On closer examination we realized our wording was not clear and conveyed the wrong meaning. We have rewritten much of the description of quantum crosspoints in terms of quantum gates and circuits. We have added a schematic representation of a quantum crosspoint in Figure 4. We have also redrawn the quantum circuit version of the fat-and-slim sparse crossbar concentrator in Figure 6 (b) using this schematic representation to make it clear that the corresponding quantum circuit is a standar quantum circuit. The relevant discussion in the previous version of the paper has been revised to reflect these changes.

Comment 3. Some typos are found in the manuscript. For example, Lines 15-16 on Page 8 miss the "|>", and Line 21 on Page 8 with wrong inequality direction. The index in the example in Lines 50-51 on Page 10 is not consistent with the definition in Lines 31-32 on Page 10. Line 50 on Page 16: input 4 should be "input 1".

Response: We thank the reviewer for the careful reading of the manuscript. We have corrected these errors, and additionally we went over the entire paper and fixed other typographical errors too.

Comment 4. Theorem 2 should be proved since it is frequently referred afterwards.

Response: We thank the reviewer for this suggestion. A proof for this theorem has been added.

Comment 5. The organization of the paper can be re-planned. Section 3 should be placed before Section 2.4 since without the concept of classical concentrator it is hard to understand Section 2.4. Proof in Sections 5.2 and 5.3 should be reduced or removed to the Appendix to make the context flow fluent. Conclusion section is too long and the complexity analysis should not be placed here.

Response: We thank the reviewer for improving the readability and organization of our paper. The proofs have been moved to appendices at the end of the paper. The introduction to classical concentrators has been moved before definitions of quantum packets and concentration assignments. Complexity analysis has been moved out of the conclusion section into a separate section.