An Efficient Fault-Tolerant Valve-Based Microfluidic Routing Fabric for Droplet Barcoding in Single-Cell Analysis

Yasamin Moradi, *Student Member, IEEE,* Mohamed Ibrahim, *Member, IEEE,*
Krishnendu Chakrabarty, *Fellow, IEEE,* and Ulf Schlichtmann, *Member, IEEE*

***Abstract*—Single-cell analysis is used to gain insights into diseases such as cancer. Advances in microfluidic solutions have enabled the efficient classification and analysis of a heterogeneous population of cells. Recently, a hybrid microfluidic platform was proposed for concurrent single-cell analysis on thousands of heterogeneous cells. In this design, barcoding droplets are routed using a valve-based routing fabric to label the input cells. However, prior work overlooked defects that are likely to occur during chip fabrication and system integration and the fault tolerance of this routing fabric remains a major concern. We address the above limitation and introduce a low-overhead design technique for guaranteeing the tolerance of single faults, while maintaining the efficiency of the cell-analysis platform. We show that the proposed method is optimal in that it minimizes the overhead in terms of fabric size.**

***Index Terms*—Biochips, valve-based microfluidics, routing fabric, fault tolerance, droplet barcoding.**

I. Introduction

SINGLE-CELL analysis is used to advance our understanding of diseases such as cancer [1]. The flow of a single-cell analysis experiment consists of three steps—namely cell encapsulation and differentiation [2], droplet barcoding [3], and type-driven cell analysis [4], [5]. Using recent advances in microfluidic technologies, thousands of heterogeneous cells can now be concurrently analyzed in a high-throughput manner. These miniaturized platforms are typically based on two distinct microfluidic technologies. Flow-based microfluidic platforms consist of microvalves that are controlled via pneumatic inputs, and channels that are used to guide liquid flow [6], [7]. On the other hand, in digital microfluidic biochips (DMFBs), a two-dimensional array of electrodes is used to manipulate droplets [8].

Microfluidic techniques have recently been developed to perform each step of the single-cell analysis flow. However, each of these steps can only be carried out efficiently in a specific microfluidic technology domain. For example, flow-based platforms are more effective than DMFBs for interfacing to the external world [9]. In addition, cell encapsulation can be performed rapidly on flow-based platforms [4], [10], while DMFBs enable real-time decision making and are more suitable for sample processing [11], [12], [13], [14]. Therefore, to perform single-cell analysis efficiently on a single chip, a platform integrating these two domains is desirable.

A hybrid platform for single-cell analysis was recently introduced in [3]. This platform consists of components that work in different microfluidic domains and are connected to each other by suitable interfaces. Each of these components is designed to conduct a specific step of the single-cell analysis. In this platform, a valve-based routing fabric is used to perform droplet barcoding. To keep track of the identity of cells in a single-cell analysis flow, each cell most be labeled using a distinct barcode that corresponds to the cell's type. A single-cell experiment may require hundreds of distinct barcodes [15]. The routing fabric is utilized as a crossbar to route barcoding droplets from the reservoirs that are connected to its input ports, to the digital microfluidic part that is connected to its output ports. The barcoding droplets are then mixed with samples on the digital microfluidic part [5].

To construct the routing fabric, transposers are utilized and they are connected to each other using channels [16]. Fig. 1 illustrates the layouts and models of full and half transposers. Each input is connected to both outputs of the full transposer through valves and channels. For example, in the full transposer shown in Fig. 1(a), when V_1 and V_6 are closed but V_2 and V_5 are open, $input_1$ is connected to $output_2$. A half transposer has only one output but it can be connected to either input. The connections between the input ports and output ports of the transposers are represented by straight and crossed lines in the models. Fig. 2 shows the 3D representation of a full transposer and its two different configurations. Fig. 3 shows the layout of a 6-to-4 crossbar. As shown in this figure, transposers are connected to each other using channels. A path for connecting i_1 to o_4 and a path for connecting i_6 to o_2 are also shown in Fig. 3. The configuration of the transposers and the location of the barcoding droplets at four different time steps are shown in Fig. 4. In this example, two barcoding droplets enter the crossbar at the same time from i_1 and i_6. These droplets are then guided to the requested outputs. Fig. 6 shows the models of a 2-to-2 crossbar and an 8-to-4 crossbar.

It was shown in [3] that this crossbar enables scalable and high-throughput droplet barcoding and it has a smaller footprint compared to the alternative of connecting reservoirs directly to the digital microfluidic part. However, the fault tolerance of the crossbar is a major concern for this design. Recent studies show that channels and valves may be blocked due to physical defects that can occur during fabrication [17], [18]. Moreover, valves operate reliably only for a limited num-

Preliminary versions of this paper were presented in the 2018 Design, Automation and Test in Europe (DATE) Conference (four pages) and the 2018 European Test Symposium (two pages).

Y. Moradi and U. Schlichtmann are with the Chair of Electronic Design Automation, Technical University of Munich, 80333 Munich, Germany (E-mail: {yasamin.moradi, ulf.schlichtmann}@tum.de).

M. Ibrahim and K. Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, 27708, USA (E-mail: {mohamed.s.ibrahim, krishnendu.chakrabarty}@duke.edu).

Fig. 1: (a) Layout of a full transposer; (b) layout of a half transposer; (c) model of a full transposer; (d) model of a half transposer.

Fig. 2: 3D Layout of a full transposer and its two different configurations: (a) straight; (b) crossed.

ber of actuations [19]. Connections in transposers are prone to failures since they are composed of channels and valves. The failure of a connection along the path between an input and an output of the crossbar will imply that the barcoding droplet directed along this path will not pass through the connection. If no alternative path exists between this input/output pair, a sample in the downstream part will not receive any barcoding droplet. Therefore, no meaningful conclusion will be drawn from the analysis of the sample, as there will be no information on its identity.

The original routing fabric from [3] ensures connectivity for every input and output pair, but it does not guarantee path redundancy for each input/output pair. In this work, we address the fault tolerance of the routing fabric with respect to the failure of connections in transposers. We present the design of an efficient fault-tolerant crossbar and show that the redesigned crossbar ensures the existence of multiple paths through different transposers for each input/output pair, while maintaining full connectivity as defined in [3]. Post-fabrication tests can detect faulty connections. In addition, faulty connections can be detected at run time by using sensors at the outputs of the crossbar. A path with a faulty connection can be avoided by using a dynamic routing algorithm and a barcoding droplet can be routed through an alternative path. As a result, each sample in the downstream part receives a barcoding droplet.

If a defect is detected during a post-fabrication testing procedure, the faulty chip must be discarded. However, we show in this work that disposal of a chip can be avoided at low cost. More importantly, if the platform fails during an experiment, the experiment must be repeated on a new chip, i.e., more reagents will be used and the sample must

Fig. 3: Layout of a 6-to-4 crossbar and the paths connecting i_1 to o_4 and i_6 to o_2.

be obtained again. Repeating an experiment also includes the additional effort of sample preparation [20]. Moreover, obtaining a sample again may be difficult or impossible. For example, in cancer genomics, cellular heterogeneity is studied among cells obtained from a tumor [1]. A treatment developed for one cell population may not be effective against other cell populations [21]. Therefore, it is important to understand cellular heterogeneity in the target tumor. If the sample from the tumor cannot be obtained again, the experiment cannot be repeated and the opportunity to identify rare cell types will be lost. Additionally, commonly used reagents are very expensive. For example, the cost of some antibodies and the cost of Horseradish peroxidase (HPR), which is often used to determine the presence of a molecular target, is estimated to be around $8000/gram [22], which is comparable to the cost of diamond [23]. Therefore, repeating an experiment can significantly increase the cost of experiment, and this undesirable outcome can be avoided by the use of a fault-tolerant platform.

In this work we have focused only on creating alternative paths and left testing and dynamic routing to future work. The key contributions of this paper are as follows:

- We define the concept of critical input/output pairs and present a method to identify them.
- We develop a generalized theory of crossbar fault tolerance and introduce a method for expanding a given crossbar to make it fault-tolerant.
- We show that our method adds minimum area overhead to the design.
- We present a comprehensive performance evaluation and assessment of fault tolerance for the expanded crossbar.

The rest of this paper is organized as follows. In Section II, we explain the single-cell analysis flow and describe the architecture of the platform. In Section III, we introduce the fault model that is considered in this work, and the defects that result in such faults. In Section IV, we present a mathematical model for the fault-tolerant crossbar design problem. Next, we present a method for identifying the input/output pairs that may become disconnected in the case that a failure occurs (Section V). In Section VI, we present a new method for expanding the crossbar and describe the design of an efficient fault-tolerant valve-based routing fabric. We present formal

Fig. 4: Time-lapsed image of a 6-to-4 crossbar. A barcoding droplet is transferred from i_1 to o_4. At the same time, another barcoding droplet is transferred from i_6 to o_2.

proofs to characterize the fault tolerance of this new crossbar design. Comprehensive performance evaluation, overhead comparison, and fault-tolerance assessment are presented in Section VII. Finally, Section VIII concludes the paper.

II. Single-Cell Analysis

In a single-cell analysis procedure, thousands of heterogeneous cells are explored and analyzed individually in order to understand the cell population. This procedure enables researchers to link molecular events in a single cell to the behaviour of the cell population. Single-cell flow consists of several steps. The first step towards single-cell analysis is isolating input cells and encapsulating them inside droplets [24]. Next, heterogeneous cells are differentiated according to their type. Cell differentiation can be done by using different methods, based on different signaling events in cells; e.g., fluorescence intensity from a fluorescent protein [25], [26], [2], [3]. When a sample cell enters the platform and its type is identified, a barcoding droplet that represents the cell type must be dispensed and mixed with the sample. Droplet barcoding is essential to keep track of the samples, because in the next step, each cell is manipulated through biochemical procedures and its type may no longer be identifiable. Based on the cell type, a single-cell bioassay such as qPCR [11] is carried out on each cell. The result of type-driven cell analysis is associated with droplet barcodes [5]. This enables the drawing of meaningful conclusions from the experiment.

Each step of the single-cell analysis is carried out in a specific component of the platform introduced in [3]. Fig. 5 shows a model of this platform. When an encapsulated cell enters the platform, its type is identified by using fluorescent detectors in the DMFB. Next, a suitable barcoding droplet is dispensed from one of the reservoirs, each of which contains one type of barcoding droplets. A routing fabric with n input ports and m output ports (as shown in Fig. 6(b)) connects the reservoirs with the DMFB and is used to direct barcoding droplets from the reservoirs to the DMFB. Samples are then mixed with barcoding droplets in the DMFB and single-cell bioassays are performed according to the cell type.

As shown in Fig. 6, the routing fabric that is used for droplet barcoding consists of interconnected transposers. A selection of transposers can be activated to direct a droplet from a reservoir to one of the output ports. The droplet is guided through a path, which consists of channels and valves. The failure of a channel or a valve will imply that the droplet will not pass through the defective part and will be stuck in the routing fabric. As a result, a sample in the downstream DMFB will not receive any barcoding droplet. This means that no meaningful conclusion can be drawn from the analysis of the sample. However, if other channels and valves can be used for directing the barcoding droplet from the reservoir to the desired output port of the routing fabric, another barcoding droplet of the same type can be dispensed and delivered to the sample.

Fig. 5: The hybrid platform for single-cell analysis.

For example, it was shown in [27] that single-cell analysis can be used to study the cellular composition of colon cancer epithelia. For this purpose, first, cells must be sorted and barcoded according to their cell types. Next, single-cell PCR is used to distinguish gene sets amongst cellular populations contained in normal human tissues and cancer epithelia. Traditional assays provide information on average gene expression. The use of microfluidic technologies allows parallel analysis of the gene expressions for each cell and opens the possibility of analyzing hundreds of cells in parallel. The large number of measurements, together with the barcoding mechanism, allows statistical analysis of the cell population. Cells can be clustered into groups with similar gene expression profiles. Colon epithelium contains heterogeneous populations of cells. Based on their lineage, differentiation stage and functional status, these cell populations express different protein markers that can be identified during the experiment. It was shown in [27] that gene-expressions and lineage differentiation, can largely explain the transcriptional diversity of cancer tissues and are strongly associated with patient treatment. This experiment can be performed on the hybrid platform. However, if an input/output pair of the crossbar are disconnected and some of the cells do not receive any barcodes, their types cannot be identified after the analysis. As a result, no meaningful conclusion can be drawn from the clustering of these cells. Therefore, our goal is to guarantee the existence of multiple paths between every input/output pair of the routing fabric.

III. Fault Model

Despite their tremendous potential for biochemical analysis, flow-based microfluidic biochips are vulnerable to defects. These defects can be associated with the fabrication steps, especially when soft lithography techniques are used [28], or they may be caused by environmental factors such as pollutants or an imperfect wafer mold. It has been shown that environmental parameters during fabrication, such as humidity, have an impact on some of the physical properties of the valves [18]. One of the main points of failure is the valve membrane, which can collapse and bond irreversibly to the flow channel or the control channel. In addition, the growing degree of integration increases the probability of defective chips. If a platform is not designed to be fault-tolerant, the experiment that is carried out on this platform must be repeated on a new chip if errors occur during the experiment. Repeating a single-cell analysis procedure is expensive, if not impossible, because samples are hard to obtain.

Fig. 6: (a) A 2-to-2 crossbar (full transposer) and its corresponding graph $F_{2 \times 2}$; (b) An 8-to-4 crossbar; (c) corresponding $F_{8 \times 4}$ for 8-to-4 crossbar.

Some of the most common defects that can occur in flow-based biochips are disconnected or blocked channels, misalignment between the control layer and the flow layer, faulty pumps that fail to generate pressure, and degradation of valves [17]. It was shown in [29] that the disconnection or blockage of a channel may be caused by the existence of environmental particles or an imperfect mold. It was also shown in [29] that if control layer and flow layer are misaligned, membrane valves cannot be closed or may not be even formed. Moreover, if a pump is faulty, it cannot generate pressure when actuated. The consequence of all these defects in biochips can be described as a block or a leak [29]. The corresponding faulty behaviour associated with the defects mentioned above is that liquid cannot pass through the faulty channel or valve. As illustrated in Fig. 1, each input is connected to all outputs of the transposer through valves and channels. We consider a fault model in which a combination of defective valves and channels block the flow of liquid along one path. Such a fault leads to the disconnection of an input/output pair inside a transposer. For example, in Fig. 1(a), a fault in V_2, V_5, or the channel between these two valves will disconnect input$_1$ from output$_2$.

Note that only defects that result in blocked connections are addressed in this paper. Other potential defects such as defective spots on the wall that can connect neighboring channels and errors in physical dimensions that can lead to leakage are not considered in this fault model. This work can be extended by considering other types of defects that result in the leakage of fluid in the flow layer or the control layer as described in [17].

IV. Problem Formulation

It was shown in [3] that a crossbar with n inputs and m outputs can be mapped to a directed acyclic graph (DAG) $F_{n \times m} = (V_{n \times m}, E_{n \times m})$. Each vertex $v \in V_{n \times m}$ represents an input or an output of a transposer. Each edge $e \in E_{n \times m}$ represents a connection inside a transposer. As an example, $F_{2 \times 2}$ represents a full transposer, as shown in Fig. 6(a). Fig. 6(b) shows an 8-to-4 crossbar, which can be mapped to the graph model shown in Fig. 6(c). It was shown in [3] that $F_{n \times m}$ is a fully connected crossbar if $F_{n \times m}$ has q vertical levels, where $q = \frac{m+n}{2}$, and n and m are even integers. In this crossbar, n nodes reside at each of the first w_1 vertical levels, where $w_1 = m - 1$. This number is decreased by two at each of the subsequent w_2 vertical levels, where $w_2 = \frac{n-m}{2}$, until it reaches m. The last vertical level consists of m nodes, which represent the output ports. We address each node in the crossbar with its coordinates. In other words, a node $p \in V_{n \times m}$ is represented as a two-tuple of the following form: (vertical coordinate of p, horizontal coordinate of p). The vertical coordinate and the horizontal coordinate of a node represent the vertical level and the horizontal level at which the node is located, respectively. An input i_k refers to the node $(1, k)$ in the crossbar, where $1 \leq k \leq n$. Similarly, an output o_j refers to the node $(q, j + \frac{n-m}{2})$, where $1 \leq j \leq m$.

The existence of a path from an input i_k to an output o_j depends on the functionality of the edges that connect the intermediate nodes. An edge whose failure (i.e., a connection failure in the transposer) results in the disconnection of an input/output pair is referred to as a *critical edge*. An edge can only be critical if it is a part of all the paths between an input/output pair. To maintain full connectivity in the case of an edge failure, the crossbar should not have any critical edges. In other words, to ensure fault-tolerance of the crossbar, every input/output pair must be connected through at least two non-overlapping paths, i.e., paths through different edges.

This problem is very similar to that of investigating the connectivity of a flow network, in which the capacity of each edge is one [30]. Such connectivity problems have attracted a lot of attention in the past because of the variety of applications in which they arise. For example, the Ford-Fulkerson method can be used for finding the minimum number of edges whose failure can disconnect a given graph [30]. An implementation of this method was presented in [31]. An efficient solution for finding all minimum-sized sets of critical edges in a graph was presented in [32]. However, the use of such methods to identify critical edges requires additional effort for case-by-case analysis of each crossbar and careful redesign based on this analysis.

A general method is needed for achieving fault tolerance with less effort and during the design of the crossbar. In addition, an irregular crossing of edges cannot physically be implemented using transposers, because transposers support the crossing of only two edges. For example, if a node in the crossbar is required to be connected to more than two nodes as shown in Fig. 7(a), the edge that will be created for making this connection will cross two existing edges. As shown in Fig. 1, the layout of a transposer is designed to enable crossing of only two edges, i.e., two channels etched in different layers. The replication of an edge introduces additional crossings, which implies that the redesigned crossbar cannot be implemented using only transposers. As shown in Fig. 7(b), the use of previous methods requires an additional component that enables crossing of more than two edges and a new physical design method to generate the layout of such a crossbar. To guarantee the fault tolerance of the crossbar using existing components, a method is needed that considers the unique structure of the crossbar.

Fig. 7: (a) A node in the crossbar with connection to three nodes; (b) model of a component required for implementing crossing of three edges in the crossbar.

If an input/output pair in the crossbar is connected through only one path, the failure of any edge in this path results in disconnection of the input/output pair. We refer to such an input/output pair as a *critical input/output pair*. Our goal is to redesign the crossbar in such a way that each of its inputs is connected to each of its outputs through at least two non-overlapping paths.

V. Identifying Critical Input/Output Pairs and Critical Edges

To investigate the number of non-overlapping paths between an input/output pair, we examine the connection between this input/output and the nodes at vertical level w_1. This vertical level is specifically interesting since the number of nodes in the vertical levels is reduced from this level onwards. Note that each node in $F_{n \times m}$ is only connected to the nodes in the previous and next vertical level. In order to reach an output port from an input port, none of the intermediate vertical levels can be skipped. Therefore, any path between an input/output pair goes through a node at vertical level w_1.

If an input i of a transposer T is connected to a node p further downstream in the crossbar, any path from i to p goes through one of the outputs (o) of T. Because both inputs of any full transposer are connected to both of its outputs, the other input of T is also connected to p through o. A similar argument holds for the connection between the outputs of a transposer and nodes further upstream in the crossbar. Note that i_{k-1} and i_k are inputs of the same transposer if $k = 2K$, where $2 \leq 2K \leq n$. Similarly, o_{j-1} and o_j are outputs of the same transposer if $j = 2J$, where $2 \leq 2J \leq m$.

Let us define R_k as the set of nodes at vertical level w_1 that are connected to i_{k-1} and i_k, and S_j as the set of nodes at

Fig. 8: The sets R_4, S_4, and $P_{4,4}$ in an 8-to-4 crossbar.

this vertical level that are connected to o_{j-1} and o_j. Let also $P_{k,j}$ be the intersection of these two sets, i.e., $P_{k,j} = R_k \cap S_j$. For example, Fig. 8 illustrates R_4, S_4, and their intersection $P_{4,4}$ in an 8-to-4 crossbar. The red and the green dotted lines in Fig. 8 show the nodes that can be used to connect i_4 to the nodes in R_4 and o_4 to the nodes in S_4, respectively. All the paths from i_k to o_j go through a node in $P_{k,j}$ while any other node at vertical level w_1 is either not connected to i_k or not connected to o_j. Since the crossbar is guaranteed to be fully connected [3], $P_{k,j}$ can never be empty.

To identify the nodes in R_k and S_j, we present the following two lemmas. Note that, without loss of generality, k and j are assumed to be even numbers and proofs are provided for i_{k-1} and i_k or o_{j-1} and o_j.

Lemma 1. At vertical level v, where $v \le w_1$, i_{k-1} and i_k are connected only to the nodes in $\{(v, \max\{1, k-v+1\}), \ldots, (v, \min\{n, k+v-2\})\}$.

Proof: We use proof by induction to investigate the connectivity of the nodes in the crossbar.

Base Case: We start with the first vertical level located immediately after the inputs of the crossbar, i.e., vertical level 2. Inputs i_{k-1} and i_k are connected only to nodes $(2, k-1)$ and $(2, k)$ at vertical level 2 through one full transposer.

Induction Step: Next, we study the connections between vertical levels further in the downstream. Assume that inputs i_{k-1} and i_k are connected only to the nodes in $\{(L, H_1), \ldots, (L, H_2)\}$ at vertical level L. In this case, as shown in Fig. 9(a), i_{k-1} and i_k are connected only to the nodes in $\{(L+1, H_1-1), \ldots, (L+1, H_2+1)\}$ at vertical level $L+1$. This is true because all the nodes at vertical level L are connected to the nodes with the same horizontal level at vertical level $L+1$. In addition, there is exactly one connection from node $p_1 = (L, H_1)$ to node $p_3 = (L+1, H_1-1)$ and there is exactly one connection from node $p_2 = (L, H_2)$ to node $p_4 = (L+1, H_2+1)$. No other node at vertical level $L+1$ is connected to the nodes in $\{(L, H_1), \ldots, (L, H_2)\}$.

It is obtained by mathematical induction that since inputs i_{k-1} and i_k are connected only to the nodes in $\{(2, k-1), (2, k)\}$ at vertical level 2, they are connected only to the nodes in $\{(2+(v-2), (k-1)-(v-2)), \ldots, (2+(v-2), k+(v-2))\} = \{(v, k-v+1), \ldots, (v, k+v-2)\}$ at vertical level v. However, since the routing fabric has only n horizontal levels at vertical level v, i_{k-1} and i_k are connected only to the nodes whose horizontal coordinate do not exceed the horizontal level limits, which corresponds to the nodes in $\{(v, \max\{1, k-v+1\}), \ldots, (v, \min\{n, k+v-2\})\}$.

Fig. 9: Connections between (a) the inputs and nodes at vertical level $L+1$; (b) the outputs and nodes at vertical level $L-1$.

Lemma 2. At vertical level $q-w_2-1$, o_{j-1} and o_j are connected only to the nodes in $\{(q-w_2-1, j-1), \ldots, (q-w_2-1, j+2w_2)\}$.

Proof: Similar to the proof of Lemma 1, it can be shown that if o_{j-1} and o_j are connected only to the nodes in $\{(L, H_1), \ldots, (L, H_2)\}$ at vertical level L, then they are connected only to the nodes in $\{(L-1, H_1-1), \ldots, (L-1, H_2+1)\}$ at vertical level $L-1$; see Fig. 9(b). Note that o_{j-1} and o_j are located at horizontal levels $j+w_2-1$ and $j+w_2$, respectively. Since o_{j-1} and o_j are connected only to the nodes $(q-1, j+w_2-1)$ and $(q-1, j+w_2)$ at vertical level $q-1$ through one full transposer, similar to the proof of Lemma 1, it can be proved by induction on the vertical coordinate, that o_{j-1} and o_j are only connected to the nodes in $\{(q-w_2-1, j-1), \ldots, (q-w_2-1, j+2w_2)\}$ at vertical level $q-w_2-1$.

Using Lemmas 1-2, we obtain the following key lemma for determining the number of nodes in $P_{k,j}$. Note that for even numbers k and j, $P_{k,j}$ represents the set of nodes at vertical level w_1 that can be used to connect each of the inputs i_{k-1} or i_k to each of the outputs o_{j-1} or o_j. In other words, the sets $P_{k-1,j-1}$, $P_{k-1,j}$, $P_{k,j-1}$, and $P_{k,j}$ consists of the same elements. Therefore, without loss of generality, k and j are assumed to be even numbers.

Lemma 3. $P_{k,j}$ has three or more elements for all ordered pairs (k, j) except for $(2, m)$ and $(n, 2)$. For the ordered pairs $(2, m)$ and $(n, 2)$, $P_{k,j}$ has only one element.

Proof: Let A_k and B_k be the nodes with the smallest and the largest horizontal coordinate in R_k, respectively, and let A_k^0 and B_k^0 be the nodes with the smallest and the largest horizontal coordinate at vertical level w_1 that could be connected to i_{k-1} and i_k if there was no limit on the number of horizontal levels in the crossbar. Also, let C_j and D_j be the nodes with the smallest and the largest horizontal coordinate in S_j. For simplicity, we refer to the horizontal level of the nodes A_k, B_k, C_j, and D_j as a_k, b_k, c_j, and d_j, respectively. Based on

Fig. 10: (a) $P_{2;4}$ with only one member in an 8-to-4 crossbar; (b) $P_{4;4}$ with three members in an 8-to-4 crossbar.

the crossbar structure and the above definitions, it is obtained that $1 \quad a_k \quad n \quad m + 2$, $m \quad 1 \quad b_k \quad n$, $1 \quad c_j \quad m \quad 1$, and $n \quad m + 2 \quad d_j \quad n$.

Based the above inequalities, it can be seen that $c_j \quad b_k$, and $a_k \quad d_j$. Note that $a_k \quad b_k$ and $c_j \quad d_j$. Therefore, $P_{k;j}$ has exactly one element only when either $c_j = b_k$ or $a_k = d_j$, i.e., either $k = 2$ and $j = m$, or $k = n$ and $j = 2$. In this case, the single member of $P_{k;j}$ is either $A^0_k = D_j$ or $B^0_k = C_j$.

Next, we consider the case where $P_{k;j}$ has more than one element. Because j and k always change in steps of two, A_k, B_k, C_j, and D_j move horizontally in steps of two as well. Therefore, by changing j or k, two nodes are added to $P_{k;j}$. As a result, if $P_{k;j}$ has more than one element, it has three elements or more.

For example, Fig. 10(a) shows the position of A^0_2 and B^0_2, as well as C_4 and D_4 in an 8-to-4 crossbar. Note that in this crossbar, $m = 4$. Therefore, $P_{2;4}$ is expected to have only one element. As shown in Fig. 10(a), the only node at vertical level w_1 that is connected to both i_2 and o_4, is B^0_2. Note that B^0_2 represents the same node as C_4. Fig. 10(b) presents an example for the case where $P_{k;j}$ has more than one element. It can be seen on this figure that for the input/output pair i_4 and o_4, at vertical level w_1, three nodes reside between C_4 and B_4. Therefore, $P_{4;4}$ has three elements.

We also present the following lemma for determining the number of paths between an input/output pair based on the size of $P_{k;j}$.

Fig. 11: Two non-overlapping paths between i_k and o_j that pass through p and $p + 2$ at vertical level w_1.

Lemma 4. If $P_{k;j}$ has only one element, then there exists only one path for connecting the input i_k to the output o_j. If $P_{k;j}$ has three or more elements, then i_k and o_j can be connected through at least two non-overlapping paths.

Proof: Note that since the crossbar is guaranteed to be fully connected, $P_{k;j}$ can never be empty. In the proof of Lemma 3, we showed that if $P_{k;j}$ has only one element, this element is either $A^0_k = D_j$ or $B^0_k = C_j$. At each vertical level, there is only one edge that can be used to connect each of the inputs $i_{k \quad 1}$ or i_k to each of the nodes A^0_k and B^0_k, implying that there is only one path from each of these inputs to each of the nodes A^0_k and B^0_k. Similarly, there is only one path from each of the nodes C_j or D_j to each of the outputs $o_{j \quad 1}$ or o_j. Therefore, if $P_{k;j}$ has only one element, then there is only one path for connecting each of the inputs $i_{k \quad 1}$ or i_k to each of the outputs $o_{j \quad 1}$ or o_j, which goes through either A^0_k or B^0_k.

Next, we consider the case where $P_{k;j}$ has three or more elements. Let p and $p + 2$ be two nodes at a vertical level with a difference of two in horizontal coordinates. Based on the structure of the crossbar, these two nodes have equivalent connections to the previous vertical level since they are the same outputs of two different transposers. Hence, for any two nodes p and $p + 2$ in $P_{k;j}$, we can find two non-overlapping paths that connect each of these nodes to each of the nodes $(2; k \quad 1)$ or $(2; k)$, by choosing two equivalent edges at each vertical level. Since $i_{k \quad 1}$ or i_k are connected to $(2; k \quad 1)$ and $(2; k)$ with different edges in a transposer, there exist two non-overlapping paths from each of these inputs to each of the nodes p or $p + 2$. The same is true for the paths from each of the nodes p or $p + 2$ in $P_{k;j}$ to each of the outputs $o_{j \quad 1}$ or o_j. Therefore, if $P_{k;j}$ has more than one element, there exist two non-overlapping paths from each of the inputs $i_{k \quad 1}$ or i_k to each of the outputs $o_{j \quad 1}$ or o_j. An example for this case is shown in Fig. 11. The two non-overlapping paths that connect i_k to o_j are shown in different colors.

Fig. 12(a) shows the only possible path for connecting the critical input/output pair i_2 and o_4, which goes through B^0_2. However, for connecting other input/output pairs, which are not critical, two non-overlapping paths can be found. For example, it can be seen in Fig. 12(b) that i_4 and o_4 can be connected through two non-overlapping paths.

Based on the aforementioned relation between $P_{k;j}$ and the number of non-overlapping paths between i_k and o_j, fault tolerance can be ensured by increasing the number of elements in $P_{k;j}$ for the critical input/output pairs. Therefore, our goal

Fig. 12: (a) The only possible path for connecting i_2 and o_4 in an 8-to-4 crossbar; (b) two non-overlapping paths between i_4 and o_4 in an 8-to-4 crossbar.

is to identify critical input/output pairs and find a method for connecting them to more nodes at vertical level w_1.

We refer to i_1 and i_2 as the first two inputs, i_{n-1} and i_n as the last two inputs, o_1 and o_2 as the first two outputs, and o_{m-1} and o_m as the last two outputs. Lemmas 3-4 show that the only critical input/output pairs in a crossbar are the ordered pairs (each of the first two inputs, each of the last two outputs) and (each of the last two inputs, each of the first two outputs). Using these lemmas, we obtain the following theorem to identify the critical edges.

Theorem 1. The only critical edges in a given crossbar are those that are used to connect each of the first two inputs to each of the last two outputs, or each of the last two inputs to each of the first two outputs.

Proof: Lemma 4 shows that if $P_{k;j}$ has more than one element, there exist two non-overlapping paths from each of the inputs i_{k-1} or i_k to each of the outputs o_{j-1} or o_j. Therefore, if an edge in one of these paths fails, the target input/output pair will remain connected through the alternative path. However, if $P_{k;j}$ has only one element, there is only one path for connecting each of the inputs i_{k-1} or i_k to each of the outputs o_{j-1} or o_j. Therefore, failure of an edge in this path disconnects an input/output pair, i.e., all the edges in this path are critical edges. Based on Lemma 3, $P_{k;j}$ has only one element if either $k = 2$ and $j = m$, or $k = n$ and $j = 2$. Note that these values for k and j correspond to the first two inputs and the last two outputs, or the last two inputs and the first two outputs respectively. Therefore, the only critical edges in a crossbar are those that are used to connect each of the first two inputs to each of the last two outputs, or each of the last two inputs to each of the first two outputs.

Note that if $m = 2$, then $w_1 = 1$ and the inputs reside at vertical level w_1. In a crossbar with two output ports, each node at a vertical level is connected to the nodes at the previous vertical level with two edges, except for the first two nodes and the last two nodes at this vertical level. Therefore, there exists only one path from each of the first or last two inputs to each of the outputs.

Fig. 13: Critical edges in an 8-to-4 crossbar.

Using Theorem 1, the critical edges can easily be found in a given crossbar. These edges are positioned in two diagonal paths in the crossbar. For example, Fig 13 shows the critical edges in an 8-to-4 crossbar. These edges are shown in red. By creating alternative paths for connecting the critical input/output pairs, these edges will no longer be critical. Therefore, our goal is to find a method for connecting the critical input/output pairs to more nodes at vertical level w_1.

VI. Efficient Fault-Tolerant Crossbar Design

It was shown in the previous section that fault tolerance of a crossbar can be ensured by increasing the number of elements in $P_{k;j}$ for the critical input/output pairs in the crossbar. Our objective is to find a method for connecting critical input/output pairs to more nodes at vertical level w_1 with minimum area overhead. Area overhead in the routing fabric increases the size of the platform and therefore, the fabrication cost. More importantly, an increase in the lengths of the paths in the crossbar affects the total completion time of a single-cell experiment on this platform. The efficiency of the protocol can be impacted by the completion time of the protocol. One of the concerns in biochemical analyses is that the state of the cells may be altered after sample manipulation. Therefore, cells are preserved through the experiment by applying fixation methods [33]. However, cells can be fixed only for a limited time. For example, it is shown in [34] that the fixation time in chromatin immunoprecipitation (ChIP), which is a widely used technique for studying DNA-protein binding events [35], is sensitive to cell types and can be as low as a few minutes. To ensure high protocol efficiency, the completion time of the experiment must be less than the fixation time of the samples.

Fig. 14: (a) Intersection of R_2 and S_4 in an 8-to-4 crossbar; (b) intersection of R_2' and S_4' in an expanded 8-to-4 crossbar.

Therefore, our objective is to achieve fault-tolerance with minimum increase in the lengths of the paths in the crossbar.

Using Lemmas 1-2 from Section V, we derive the following theorem for achieving a fully connected and fault-tolerant crossbar.

Theorem 2. An n-to-m crossbar with $w_1 + w_2 + 1$ vertical levels is fault-tolerant if and only if $w_1 \geq m + 1$ and $w_2 \geq \lceil \frac{n-m}{2} \rceil$.

Proof: It can be seen from Lemmas 1-2 that $P_{k,j} = \{(w_1, \max\{1, k - w_1 + 1\}), \ldots, (w_1, \min\{n, k + w_1 - 2\})\} \cap \{(w_1, j - 1), \ldots, (w_1, j + 2 \cdot w_2)\}$. Lemma 3 show that $P_{k,j}$ has more than three elements for all ordered pairs (k, j) except $(2, m)$ and $(n, 2)$. Fig. 14(a) illustrates the overlapping of R_2 and S_4 in an 8-to-4 crossbar. The only element in $P_{2,4}$ is the node with the highest value of horizontal coordinate in R_2. However, we show that by adding two vertical levels to the crossbar and expanding the crossbar following the same structure, the number of elements in $P_{2,m}$ increases. If two vertical levels are added between the vertical levels w_1 and $w_1 + 1$ in the crossbar, the expanded crossbar has m nodes at the first w_1' vertical levels, where $w_1' = w_1 + 2$. For example, Fig. 14(b) shows an expanded 8-to-4 crossbar. The edges that represent additional transposers are shown in red.

Let us define R_k' and S_j' as the set of nodes at vertical level w_1' that are connected to i_k and o_j, respectively, and let $P_{k,j}'$ be the intersection of these two sets. Since the expanded crossbar follows the same structure as the original crossbar, Lemmas 1-2 apply to this design as well. Using Lemmas 1-2 we obtain that $R_k' = \{(w_1', \max\{1, k - w_1' + 1\}), \ldots, (w_1', \min\{n, k + w_1' - 2\})\}$ and $S_j' = \{(w_1', j - 1), \ldots, (w_1', j + 2 \cdot w_2)\}$. Since the highest value of horizontal coordinate in R_k' is increased compared to R_k, $P_{k,j}'$ has more elements compared to $P_{k,j}$. As shown in Fig. 14(b), in the expanded crossbar, i_2 and o_m are connected to three common nodes. Lemma 4 shows that if an input and an output are connected to three common nodes at vertical level w_1, then there exist two non-overlapping paths for connecting them. Therefore, in the expanded crossbar, i_2 and o_m can be connected through two non-overlapping paths. It can be shown similarly that in this design, there exist two non-overlapping paths for connecting i_n and o_2. Therefore, the expanded crossbar has no critical transposers and it can tolerate the failure of any single edge.

Next, we considet the case where $w_1 \leq m$. It was shown in Lemma 3 that if $w_1 = m - 1$, then for the ordered pairs $(2, m)$ and $(n, 2)$, $P_{k,j}$ has only one element. Furthermore, it can be obtained from Lemmas 1-2 that if $w_1 = m$, then $P_{k,j} = \{(m, \max\{1, k - m + 1\}), \ldots, (m, \min\{n, k + m - 2\})\} \cap \{(m, j - 1), \ldots, (m, j + 2 \cdot w_2)\}$. Therefore, for the ordered pairs $(2, m)$ and $(n, 2)$, $P_{k,j}$ has only two elements, which are $\{(m, m), (m, m - 1)\}$ and $\{(m, n - m + 1), (m, n - m + 2)\}$, recpectively. As a result, when $w_1 \leq m$, the critical input/output pairs cannot be connected through two non-overlapping paths. Therefore, a crossbar with the configuration introduced in [3] is fault tolerant only if $w_1 \geq m + 1$.

The following statement can be derived as a consequence of Theorem 2:

Corollary 1. To ensure fault tolerance of a crossbar with the configuration introduced in [3], at least two vertical levels must be added to the design.

Proof: Lemma 4 shows that for i_k and o_j to be connected through two non-overlapping paths, they must be connected to three common nodes at vertical level w_1. Therefore, for each critical input/output pair, two nodes must be added to $P_{k,j}$. The proof of Theorem 2 shows that for this purpose, at least two vertical levels must be added to the crossbar. Although other methods may exist for ensuring fault tolerance of a given crossbar, all solutions will add more than one vertical level to the design.

Note that the original crossbar design from [3] considers only a sufficient criterion for full connectivity, Theorem 2 provides a sufficient condition for designing a fully connected and fault-tolerant crossbar. Moreover, Corollary 1 implies that the proposed method leads to a minimum increase in the depth of the routing fabric.

Based on Theorem 2, a fault-tolerant crossbar can be designed by following the structure of the crossbar introduced in [3]. In the corresponding graph for an $n \times m$ crossbar, n nodes reside at each of the first $m + 1$ vertical levels. This number is decreased by two at each of the subsequent $\lceil \frac{n-m}{2} \rceil$ vertical levels, until it reaches m at the second-to-last vertical level. The last vertical level consists of m nodes.

Fig. 15 shows the steps of our algorithm for automatically generating the graph model corresponding to a fault-tolerant crossbar. The algorithm begins by initializing the parameters w_1 and w_2 (Line 2). Next, it generates the first vertical level that consists of input nodes (Line 3). Then, the subsequent $(w_1 - 1)$ vertical levels with n nodes are generated (Lines

TABLE I: Overhead introduced by the proposed method.

n m	20 4	20 8	20 16	40 8	40 16	40 20	40 36
Number of transposers in the original routing fabric	73	106	160	321	455	516	720
Number of transposers introduced by the proposed method (overhead)	19 (16%)	19 (17%)	19 (11%)	39 (12%)	39 (8%)	39 (7%)	39 (5%)

Fig. 15: Graph model F_n $_m$ generator

4-10). At each vertical level, the nodes are connected with appropriate nodes in the previous vertical level to represent full transposers. Note that in the following w_2 vertical levels, each vertical level consists of two fewer nodes compared to the previous vertical level. The variable s is used to determine horizontal coordinates of the first node and the last node at each vertical level in this part (Line 11). The nodes and the associated edges are generated similar to the previous vertical levels (Lines 12-19). Finally, the algorithm generates the last vertical level that consists of output nodes (Line 20-24). The worst-case computational complexity of this algorithm is $O(n^2m)$.

For example, we can construct an 8-to-4 fault-tolerant crossbar as follows. First, 8 nodes must be generated at the first vertical level. Next, 4 subsequent vertical levels must be added to the design. At each of these vertical levels, 8 nodes must be generated and connected with appropriate nodes located in the previous vertical level to represent full transposers. At vertical levels 6 and 7, 6 and 4 nodes must be generated, respectively. Each node in this part must be connected with appropriate nodes located in the previous vertical level to represent full and half transposers. As the final step, 4 output nodes must be added to the design at the last vertical level.

VII. Evaluation and Simulation Results

Our goal in this section is to evaluate the redesigned routing fabric in terms of fault tolerance and transposer overhead. We also compare the performance and quality of the fault-tolerant routing fabric with the original design reported in [3]. For our simulations, we apply CoSyn [3] to the original routing fabric from [3] and the fault-tolerant routing fabric. CoSyn is implemented using C++. The set of bioassays constituting the single-cell analysis protocol [3] were used as a benchmark, and the input cells were classified using a uniform distribution function.

A. Transposer and Area Overhead

We first calculate the cost of fault tolerance in terms of the transposer and area overhead in the redesigned crossbar. The efficient fault-tolerant crossbar design method adds only two vertical levels to the crossbar. Since in the first w_1 vertical levels of the crossbar $\frac{n}{2}$ transposers reside at each odd vertical level, and $\frac{n}{2}$ 1 transposers reside at each even vertical level, the number of additional transposers in the proposed method is n 1. Table I shows the transposer overhead for various values of n and m in the fault-tolerant routing fabric.

B. Performance Evaluation

To assess the impact of our design on the performance of the routing fabric, we compare the total completion time for the single-cell analysis protocol on two platforms that use the original design [3] and the fault-tolerant routing fabric, respectively. To ensure that the results depend only on the crossbar design, the number of resources in the other parts of the platform is assumed to be unlimited. We simulate the analysis of 100 input cells of 20 and 40 different types, using a varying number of output ports for the crossbar. As shown in Fig. 16, the fault-tolerant crossbar generally results in longer completion time due to the existence of additional transposers on the paths and an increase in the length of the paths. However, by using the redesigned crossbar, high cell-analysis throughput is still supported.

C. Design-Quality Assessment

We also investigate the quality of a platform using the fault-tolerant routing fabric by comparing it to the quality of a platform using the original routing fabric. For this purpose, we evaluate the quality of the two platforms on the basis of *cell-analysis density*, i.e., the number of single-cell experiments that can be carried out on these platforms in a given window of time and using a given number of resources in the DMFB [3]. We calculate the cell-analysis density for a time window of one minute and a set of DMFB resources of size 100 electrodes. We use the term *DMFB capacity* to refer to the fraction of input cells that can be processed simultaneously on a given set of DMFB resources [3]. Therefore, DMFB capacity changes when the number of DMFB resources are varied. For example,

TABLE II: Reliability of the crossbar in the case of double faults.

n m	8 4	12 4	12 8	16 4	16 8
Number of combinations of two edges	3828	12090	27730	28680	61776
Number of cuts of size two	72	110	156	156	210
Percentage of double faults that do not impact the full connectivity of the crossbar	98.12%	99.1%	99.44%	99.46%	99.67%

Fig. 16: Completion time using the original and the fault-tolerant routing fabric with: (a) 20 barcoding inputs, (b) 40 barcoding inputs

Fig. 17: Cell-analysis density using 40 barcoding inputs and 32 barcoding outputs.

the DMFB capacity is 1 if sufficient resources are available for processing all the cells simultaneously. A DMFB capacity of 0.5 indicates that the resources are only sufficient to process half of the cells. We change DMFB capacity by using a varying number of DMFB resources.

We simulate the analysis of 100 input cells of 40 different types using 32 barcoding outputs and various numbers of DMFB resources, and record the cell-analysis density for a time window of one minute. Fig. 17 shows the result of this simulation for the two crossbar designs. As expected, cell-analysis density decreases when DMFB capacity is increased. In general, cell-analysis density is slightly less for the fault-tolerant routing fabric since the total completion time for the same number of input cells is higher and fewer cells finish analysis in a given window of time. However, as shown in Fig. 17, the cell-analysis density of the platform based on the proposed fault-tolerant crossbar is very close to that of the original platform from [3].

D. Fault-Tolerance Assessment

Finally, we evaluate the reliability of the redesigned crossbar. The routing fabric designed in this work is only guaranteed to tolerate a single fault. We show however that this design is also resilient in most cases to multiple faults. We first look into the reliability of the crossbar in the case of two faults. Furthermore, we investigate the probability that an input/output pair gets disconnected in the case of random multiple failures. We also investigate the impact of defect distribution by evaluating the reliability of the design in the case that the locations of the faults are correlated.

To evaluate the reliability of the crossbar in the case of two failures, we find all cuts of size two in the corresponding graph model for the crossbar. A cut C in a graph is a partition of vertices into two disjoint subsets [30]. The cut-set of a cut C is the set of edges that have one endpoint in each subset. The size of a cut is the number of edges in its associated cut-set. By finding all cuts of size two in a graph, we identify the pairs of edges whose simultaneous failures result in disconnection of an input/output pair. It was shown in [36] that all minimum cuts of a directed graph can be found in polynomial time. Note that minimum cuts in the proposed crossbar design are of size two. Table II shows the number of minimum cuts in crossbars with various numbers of inputs and outputs. For example, an 8-to-4 crossbar consists of 88 edges. Therefore, there are 3828 combinations of two edges in this crossbar. However, the failure of only 72 of these combinations can result in the disconnection of an input/output pair. We therefore conclude that more than 98% of all possible double faults do not impact the full connectivity of the crossbar. It can also be seen in Table II that this number increases as the size of the crossbar is increased.

The failure of a connection can be implemented by removing its corresponding edge from the crossbar. To simulate random multiple edge failures, we remove random edges in the crossbar and investigate the full connectivity of the fabric. An injected fault is benign if it does not cause any disconnections. Therefore, the crossbar is reliable if the injected faults are

Fig. 18: Mean and standard deviation of the percentage of benign double, triple, and quadruple faults.

Fig. 19: Mean and standard deviation of the percentage of benign faults when 10 and 25 faults are randomly injected in the crossbar.

benign. We carry out the experiment on five crossbars of different sizes.

First, we inject 10000 random double faults in five crossbars of different sizes and investigate the connectivity of all the input/output pairs in each crossbar. We record the number b of benign instances of double faults (i.e., they do not cause any disconnections). The percentage of benign injected faults is computed as $\frac{b \ 100}{10000}$. We repeat the experiment 20 times and record the mean and the standard deviation of the percentage of the benign injected faults among these experiments. The same experiment is carried out by injecting triple and quadruple faults. Fig. 18 shows the mean percentage of benign multiple faults for the 20 experiments and the standard deviation among these experiments. The simulation results show that the crossbar is more than 98% reliable in all the test cases. This number is higher for the larger crossbars, since these crossbars consist of more edges and it is less likely that the faults occur on a specific group of edges that can disconnect the crossbar. As expected, the reliability of the crossbar decreases when more faults occur.

Fig. 19 shows the percentage of benign faults in the case that 10 and 25 faults are randomly injected in the crossbar. A crossbar with a significantly larger number of faults is considered to be beyond repair and should be discarded. As shown in Fig. 19, a large number of faults has a greater impact on smaller crossbars, which consist of fewer edges.

In our previous experiments, faults were assumed to be non-correlated, i.e., the locations of the defects are independent of each other. However, the probabilities that defects exist at specific locations might be inter-dependent. Spatial correlations have been widely reported in integrated circuit fabrication. If defect distributions in biochip fabrication exhibit similar behavior, the reliability of the design can be affected.

Fig. 20: Mean and standard deviation of the percentage of benign faults in the case of uniform distribution.

To investigate the impact of the defect distribution, we first evaluate the reliability of the design in the case that faults are uniformly distributed across the crossbar. In this case, a fault can occur at any edge in the crossbar, but all such faults occur with equal probability. We choose multiple edges randomly based on this probabilistic distribution. Similar to the previous experiments, we inject 10000 random faults in the crossbar and record the percentage of benign injected faults. We repeat the experiment 20 times. Fig. 20 shows the mean percentage of the benign faults among these experiments. The simulation results show that the crossbar is more than 93% reliable in the case of double, triple, or quadruple faults that are uniformly distributed across the chip. Note that in our previous experiments, faulty edges are chosen by the use of a pseudo-random generator and the fault locations are assumed to be independent. However, in our second experiment, the probabilistic distribution allows faults to be inter-dependent. The difference between the results of these experiments is related to the nature of the pseudo-random generator.

Next, we repeat similar experiments based on a normal (Gaussian) distribution of faults that considers clustering. A fault can exist at any edge in the crossbar. Let x be the event that an arbitrary edge is faulty. The probability that the edge is faulty is computed based on a normal distribution and according to the distance of this edge from the location of the peak value of the distribution. In other words, to simulate spatial correlation between the locations of the defects, we assume that faults are more likely to occur near a certain location on the crossbar. The probability of failure is less for the edges that are located at a location that is at a further distance from this location. First, we assume that the probability of failure is higher near the center of the vertical level w_1 of the crossbar. Next, we assume that faults are more likely to occur near the inputs of the crossbar. Fig. 21 show the results of these experiments for these two cases. We observe that the probability that the crossbar fails is higher when faults occur near the interface of the crossbar. The reason is that the edges that are located near the inputs are more critical than the ones located near vertical level w_1. For example, Fig. 22 shows

Fig. 21: Mean and standard deviation of the percentage of benign faults in the case of normal distribution (a) near the inputs of the crossbar, (b) near vertical level lw_1

Fig. 22: The edges that can be used to connect i_6 and o_4 in an 8-to-4 crossbar.

all edges that can be used to construct a path between the chosen input/output pair. It can be seen that at vertical levels that are closer to the input, only a few edges can be used to connect the input/output pair. If faults occur at these edges, the input/output pair is more likely to be disconnected. However, there are more edges at vertical level lw_1 of the crossbar that can be used to connect the input/output pair. Therefore, if some of these edges are faulty, there is a higher probability that a path can still be found to connect the input/output pair.

Fig. 21(b) shows that the size of the crossbar does not have a significant impact on the reliability of the crossbar when faults occur near an input. It can be seen in Fig. 22 that the edges that are close to an input can only be used to connect this input to an output. Therefore, the failure of these edges only affects the existence of a path between this input and the outputs of the crossbar. Increasing the size of the crossbar does not affect reliability, since the additional edges cannot substitute the faulty ones that are close to an input. We also observe that when faults occur near vertical level lw_1, the reliability of a larger crossbar can be less than the reliability of a smaller crossbar. For example, the reliability of a 12-to-8 crossbar is less than the reliability of a 12-to-4 crossbar. This is expected because in a 12-to-8 crossbar, vertical level lw_1 is closer to the outputs. Therefore, the edges that are located near this vertical level are more critical for constructing a path between the inputs and the outputs.

Our simulation results show that the standard deviation is less when the mean percentage of benign faults is higher. This is because the maximum deviation from the mean value cannot be more than the difference between the mean value and the maximum value, which is 100% in this case. For example, when the mean value is 98%, the deviation cannot be more than 2%. Therefore, the standard deviation is less when the reliability of the crossbar is higher.

VIII. Conclusion

We have introduced a fault-tolerant design of a microfluidic crossbar for droplet barcoding in single-cell analysis. The proposed approach is based on the addition of a small number of transposers to the crossbar. We have formally shown that this approach ensures tolerance of any single edge failure while maintaining efficiency of the design. We have also shown that the resulting design is resilient to a high percentage of multiple edge failures. We have evaluated the design in terms of transposer and area overhead, analysis completion time, and cell throughput. As part of future work, we will investigate the possibility of increasing the reliability of the platform by using other architectures for the fault-tolerant crossbar.

IX. Acknowledgement

This research was supported in part by Technical University of Munich – Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n^{0}291763, as well as by the IGSSE Project FLUIDA of Technical University of Munich.

References

[1] A. Saadatpour, S. Lai, and G. Guo, “Single-cell analysis in cancer genomics,” *Trends in Genetics*, vol. 31, no. 10, pp. 576–586, 2015.

[2] J. Chen, C. Xue, Y. Zhao, D. Chen, M.-H. Wu, and J. Wang, “Microfluidic impedance flow cytometry enabling high-throughput single-cell electrical property characterization,” *International Journal of Molecular Sciences*, vol. 16, no. 5, pp. 9804–9830, 2015.

[3] M. Ibrahim, K. Chakrabarty, and U. Schlichtmann, “CoSyn: Efficient single-cell analysis using a hybrid microfluidic platform,” in *Proc. IEEE/ACM Design, Automation, and Test in Europe Conference (DATE)*, 2017, pp. 1673–1678.