

Robust Fault-Recovery Method for MEDA Biochips Using an IJTAG Network

Zhanwei Zhong, *Student Member, IEEE*, and Krishnendu Chakrabarty, *Fellow, IEEE*

Abstract—A digital microfluidic biochip (DMFB) is an attractive platform for immunoassays, point-of-care clinical diagnostics, DNA sequencing, and other laboratory procedures in biochemistry. A recent generation of biochips uses a micro-electrode-dot-array (MEDA) architecture, which provides fine-grained controllability of droplets and seamlessly integrates microelectronics and microfluidics using CMOS technology. In order to ensure robust fluidic operations and high confidence in the outcome of biochemical experiments, chip testing, fault diagnosis and fault recovery are critical for MEDA biochips. In this paper, we present an effective fault-recovery solution based on the homogeneous structure of MEDA. Since the microelectrode cell (MCs) in a MEDA biochip are identical, we add multiplexers for reconfigurability, whereby an MC with faulty components can use the hardware resources in a neighboring MC. In addition, we use the IEEE 1687 (a.k.a. IJTAG) network to reduce the number of control signals needed for the multiplexers, and to provide flexible sub-scan chain access for the fault-recovery control flow. A comprehensive set of simulation results demonstrates the effectiveness of the proposed fault-recovery solution for MEDA biochips.

Index Terms—fault recovery, IEEE 1687, MEDA, digital microfluidics, IJTAG

I. INTRODUCTION

A digital microfluidic biochip (DMFB) is an example of a lab-on-a-chip that is capable of performing biochemical experiments. Over the past decade, DMFBs have been demonstrated for high-throughput DNA sequencing, point-of-care clinical diagnostics, and protein crystallization for drug discovery [1]–[3]. A DMFB manipulates liquids as discrete droplets of nanoliter and picoliter volumes based on the principle of electrowetting-on-dielectric (EWOD). Compared with continuous-flow biochips, which use permanently etched microchannels, a DMFB offers the advantages of simple instrumentation, flexible device geometry, reconfigurability, and ease of coupling with other technologies [2]. Illumina, a market leader in DNA sequencing, has transitioned digital microfluidics to the marketplace for sample preparation [4]. This technology has also been deployed by Genmark for infectious disease testing [5] and by Baebies to detect lysosomal enzymes in newborns [6]. These milestones highlight the emergence of DMFB technology for commercial exploitation.

However, DMFBs available in the marketplace today suffer from several limitations: (i) constraints on droplet size and the

inability to vary droplet volume in a fine-grained manner, (ii) the lack of integrated sensors for real-time detection, and (iii) the need for special fabrication processes and the associated reliability/yield concerns. To overcome the above limitations, a micro-electrode dot array (MEDA) architecture has been proposed [7]–[9]. Unlike conventional digital microfluidics, where electrodes of equal size are arranged in a regular pattern, the MEDA architecture is based on the concept of a sea-of-micro-electrodes. Each microelectrode cell (MC) consists of a microelectrode, an activation circuit, and a sensing circuit. MEDA allows microelectrodes to be dynamically grouped to perform different microfluidic operations on the chip. Prototypes of MEDA biochips have been fabricated using TSMC 0.35 μm CMOS technology [8], and these devices can use a power-supply voltage of only 3.3 V [10]. MEDA also incorporates real-time capacitive sensing on every microelectrode to detect the property and the location of a droplet. The sensing map derived in this manner allows MEDA biochips to dynamically respond to bioassay outcomes, perform real-time error recovery, and execute if-then-else protocols from biochemistry.

While the above benefits continue to drive research in MEDA biochips, fabrication defects, imperfections, and wear-out are major concerns on this emerging technology platform [11], [12]. Testing and fault tolerance are therefore key for technology maturation. According to [12], there are two typical defect types in MEDA biochips, namely intra-MC defects (e.g., transistors in the MC are stuck-on) and inter-MC defects (e.g., hard- and resistive-shorts between two microelectrodes). Defects may eventually result in errors, which can adversely impact the correctness of the entire experiment [11]. In order to ensure an adequate quality level before being used for bioassay execution, MEDA biochips need to be tested and diagnosed for defects. Moreover, many biochips are expected to be used for healthcare and medical diagnostics. Therefore, these biochips must be designed to be fault-tolerant such that they can continue to operate reliably in the presence of faults.

There has been early work on test methods and fault-tolerance strategies for MEDA biochips. A comparison between these methods are summarized in Table I. The oscillation-based test method in [13] can only detect faults in the micro-electrode layer (i.e., dielectric breakdown and charge trapping) and in the sensing circuit. The work in [11] presents a built-in-self-test (BIST) architecture that can detect faults in the scan-chain, and detect/locate faults in activation/sensing circuits.

The above two methods address fault detection and fault location, but they did not attempt to tolerate or recover from defects. Test methods for MEDA must also consider the scan-chain design, in which D flip-flops (DFFs) under the MCs are serially

A preliminary version of this paper was published in the Proceedings of ITC, 2019. This research was supported in part by the National Science Foundation under grants CCF-1702596 and ECCS-1914796.

Z. Zhong and K. Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, 27705 USA, e-mail: (zz114@duke.edu, krish@duke.edu).

TABLE I: Comparison between methods for testing MEDA biochips.

Methods	Micro Electrode	Activation Circuit	Sensing Circuit	D Flip-Flop	Interconnect
JET'17 [13]	D	✗	D	✗	✗
ITC'16 [11]	✗	D, L	D, L	D	D
ITC'18 [14]	✗	✗	✗	L, R	✗
ITC'19 [15]	✗	R	R	R	R
Proposed	✗	L, R	L, R	L, R	L, R

Note: “✗” means unavailable, “D” means fault detection, “L” means fault location, “R” means fault recovery.

connected (see Section II.A). Additionally, a diagnosis and fault-tolerant design specific to MEDA biochips was proposed in [14]. This design aims to detect and locate one or multiple faults in the DFFs, and ensures recovery from these faults. However, faults are not limited to DFFs; can also occur in the activation and sensing circuits in the MEDA platform. This method cannot detect and tolerate defects in the activation and sensing circuits. Moreover, [14] does not consider defects that lead to an open between two neighboring MCs (i.e., interconnects).

A fault-recovery method based on an IJTAG network is proposed in [15]. This method incorporates two types of circuits such that a neighboring MC can operate as a backup for fault-recovery. This solution can be used to recover activation/sensing circuits, DFFs and interconnects (between DFFs) from faults. However, this method only allows the recovery operation from a neighboring MC; if the neighboring MC is also faulty, then fault-recovery is not possible. In addition, because of the use of bypass MUX (ahead of each DFF), a scan-chain failure can also be caused by either an open-fault between MCs or a faulty bypass MUX. Therefore, a diagnosis method is needed to analyze the cause of a scan-chain failure. Finally, in the original MC design, the MC-activation circuit and the MC-sensing circuit share the same set of components. This design is not robust because if a fault occurs in either the MC-activation circuit or the MC-sensing circuit, the other one is also deemed to be faulty.

Paper Contributions

In order to overcome the drawbacks of [15], we propose a more efficient fault-recovery solution for MEDA biochips. Compared with [15], the key contributions of this paper are as follows:

- We propose a new fault-recovery design for each MC such that if an MC contains a defect, it can use the hardware resources of another MC to operate correctly. Note that in [15], fault recovery can only be implemented using a neighboring MC. If faults appear in neighboring MCs, we cannot recover from them. However, in the new fault-recovery design, we can still recover from these faults.
- We present a diagnosis method to determine whether a scan-chain failure in MEDA is caused only by faulty DFF(s) (e.g., stuck-at fault). If the scan-chain failure is caused only by faulty DFF(s), recovery can be carried out using the method

in [14]. Otherwise, the scan chain is deemed to be not usable and will be blocked.

- We present a new MC design such that the MC-activation circuit and the MC-sensing circuit are separated (i.e., there is no hardware sharing). In this case, a fault in the MC activation circuit will not affect the functionality of the MC sensing circuit, and vice versa.
- A comprehensive set of simulation results demonstrates the effectiveness of the proposed fault-recovery solution for MEDA biochips.

The remainder of the paper is organized as follows. Section II describes the basics of microelectrode cells (MCs) in MEDA and the IJTAG network. Section III describes the fault-recovery design for each MC. Section IV presents the new MC design and the corresponding test method. Section V presents the IJTAG network used in our design and discusses the control flow for fault recovery. Section VI presents simulation results to illustrate the effectiveness of the proposed technique. Finally, Section VII concludes the paper.

II. BACKGROUND

A. Microelectrode Cell

A MEDA biochip consists of repeated instances of a basic unit called the microelectrode cell (MC); see Fig. 1. A typical MC includes four parts: a microelectrode (a top plate + a bottom plate), a DFF, an activation circuit, and a sensing circuit. There are three basic operations for an MC:

1) MC activation. In order to perform droplet operations on MEDA, the biochip needs to activate a group of MCs to form a micro-component (e.g., splitter or mixer). In this operation, the control circuit set $CONT = 1$, $ACT = 1$, $IN = 1$, and a 25 V voltage is applied to the top plate [16]. If a rising edge of MC-CLK is applied to the DFF, pin Q will output logic “1”, and transistors T3 and T4 are switched on while transistors T1 and T2 are switched off. In this case, the bottom plate is directly connected to the ground (0 V). Due to the potential difference, it will generate a force that can drag a droplet towards itself.

2) MC de-activation. In this operation, the control circuit set $CONT = 1$, $ACT = 1$, $IN = 0$, and a 25 V voltage is applied to the top plate [9]. If a rising edge of MC-CLK is applied to the DFF, pin Q will output logic “0”, and transistors T1, T2, T3 and T4 are switched off. When an MC is de-activated, the high voltage of 25 V is still applied to the top plate, In this

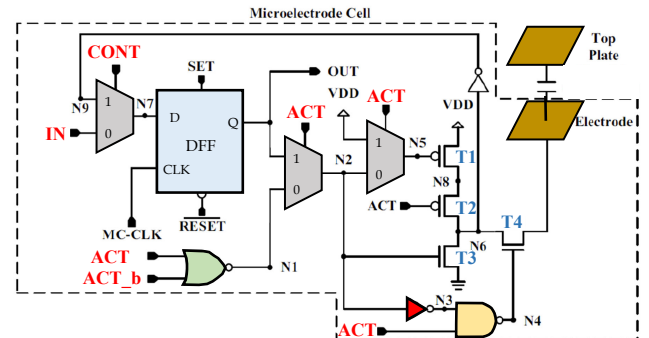


Fig. 1: Schematic of an MC in MEDA biochips.

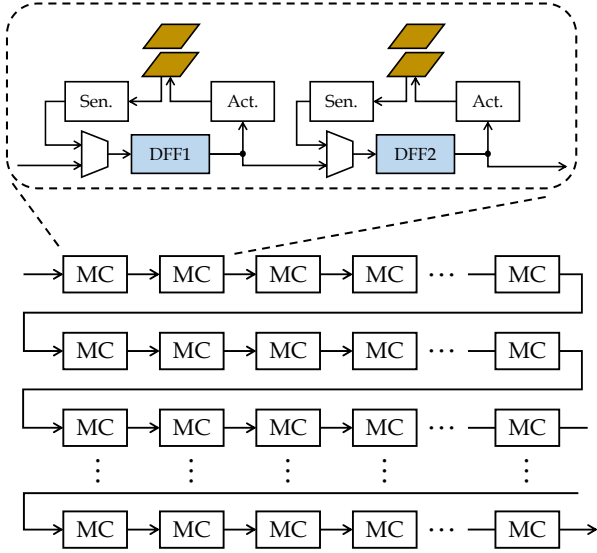


Fig. 2: An illustration of the scan-chain structure in MEDA.

case, the bottom plate is floating and an induced voltage of 17 V is generated [7]. Because the potential difference is smaller than the threshold, no force is generated.

3) MC sensing. MC sensing is used to measure the capacitance between the top plate and bottom plate. In this operation, the control circuit set $CONT = 0$, $ACT = 0$, $ACT_b = 1$, and the top plate is connected to the ground. In this case, transistors T1, T2, and T4 are switched on while transistor T3 is switched off, which charges the bottom plate to voltage 3.3 V (i.e., VDD). Next, the control circuit set $ACT_b = 0$, and transistors T1, T3 and T4 are switched on while transistor T2 is switched off. As a result, the bottom plate is now connected to the ground, and the voltage of N6 decreases due to discharging. By applying a rising edge of MC-CLK at a preset time, a value of “0” or “1” will be stored in the DFF. If a droplet is present between the top plate and bottom plate, the discharge rate is slower, and a value of “1” is stored; otherwise, the DFF will store a value of “0”.

B. Scan-Chain Design

The logic abstraction of an MC is shown in Fig. 1. The activation/de-activation status of an MC is determined by the value stored in the DFF: a value of “1” indicates MC activation; otherwise, the MC is de-activated. In the following discussion, we refer to the activation value for each MC as the *MC-activation value*.

After MC sensing, the result is written to the DFF indicating if a droplet is present or not. In the following discussion, we refer to the capacitance sensing value for each MC as the *MC-sensed value*.

In order to assign the MC-activation value to each MC and read out the MC-sensed values, all the MCs on MEDA are connected using a single scan-chain structure, i.e., the output of one MC is connected to the input of the next MC (see Fig. 2). Using this scan chain, a sequence of bits is shifted into the scan chain such that the value in each DFF can be programmed. We refer to the sequence of bits as an *activation pattern*. Likewise, after MC-sensing operation, the 0/1 MC-sensed values are stored

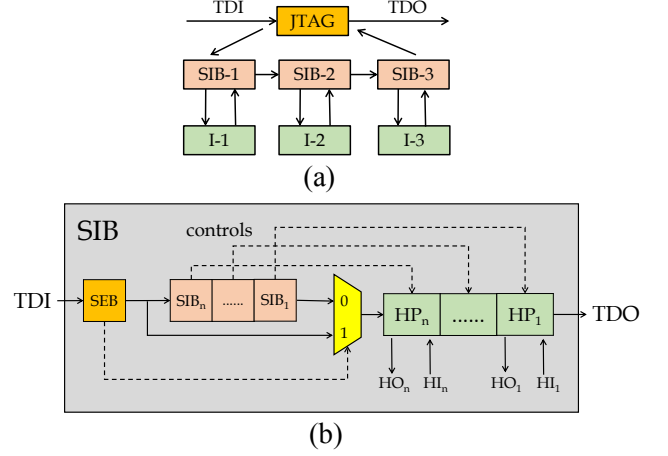


Fig. 3: (a) An example of an IJTAG network and (b) a simplified view of the SIB design.

in each DFF, MC-sensed values are obtained by shifting out a sequence of bits, which is referred to as a *sensed pattern*.

To carry out a bioassay (i.e., a bio-chemical experiment) on a MEDA biochip, we first utilize a synthesis algorithm to generate the fluidic module schedule, droplet routing and module placement for the bioassay [17], [18]. Next, these three elements are translated into a series of activation patterns. During the execution of a bioassay, an activation pattern is shifted in the MCs, and then a sensed pattern is shifted out to obtain the status of each droplet. This process is repeated until the completion of the bioassay.

C. The IJTAG Network

The IEEE Std. 1687 (a.k.a. IJTAG) [19] provides flexible access to on-chip instruments through the IEEE 1149 JTAG test access port (TAP) [20]. It is now being increasingly used for post-silicon validation, production test, fault diagnosis, and fault monitoring [21]. To provide flexibility of instrument access, a hardware component called the Segment Insertion Bit (SIB) has been introduced [22]. The SIBs in the 1687 network are used to select or unselect multiple network segments for the scan chain. It operates in two states: (1) if it is open, it includes the segment in the scan path; (2) if it is closed, it excludes the segment from the scan path. The state of the SIB is configured by first shifting in a control bit (“0” for close and “1” for open) into its register, and then updating its register on capture, shift, and update (CSU) cycles [22].

As shown in Fig. 3(a), the JTAG interface is a doorway that controls and manages SIB-1, SIB-2, and SIB-3. In addition, three instruments (e.g., sub-scan chain) I-1, I-2 and I-3 are connected to SIB-1, SIB-2, and SIB-3, respectively. Suppose we need to access the instruments I-1 and I-3, in this case, we only need to configure the control bits of SIB-1, SIB-2 and SIB-3 as “1”, “0” and “1”, respectively. As a result, I-1 and I-3 are selected while I-2 is unselected.

As shown in Fig. 3, a SIB component primarily includes three parts:

1) Hierarchical Port (HP): Each HP is connected to a lower level of the IJTAG network segment.

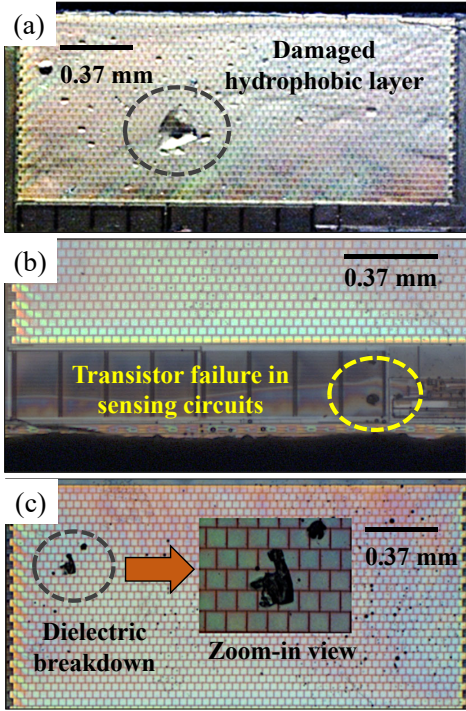


Fig. 4: Images of some typical visible defects in a fabricated MEDA biochip. (a) Defect in the hydrophobic layer. (b) Defective transistors in peripheral circuits. (c) Defect in the dielectric layer.

2) SIB Bits: When $SIB = 1(0)$, the corresponding HP is open (close), and the segment on this HP is included (excluded) in (from) the primary scan path.

3) SIB Exclusion Bit (*SEB*): When $SEB = 1$, the SIB bits are not included into the scan path (i.e., they are bypassed). This feature can be utilized to reduce the SIB overhead [23]. When $SEB = 0$, the SIB components are included in the primary scan path.

D. Defects in MEDA

As feature sizes scale down in MEDA biochips, the sizes of microelectrodes and distances between microelectrodes are also reduced to achieve higher levels of integration. This increasing density increases the likelihood of defects. Some typical defects and the corresponding fault models are listed below:

1) Defects in the hydrophobic layer. A hydrophobic layer is used to increase the electrowetting force for transporting droplets [2]. This layer can be damaged by chemical reactions and physical scratches. A damaged hydrophobic layer, as shown in Fig. 4(a) for a fabricated chip, cannot provide adequate electrowetting force when electrodes are actuated.

2) Transistor failures. These failures in control and sensing circuits can result in incorrect droplet actuation, maintenance, and sensing. An example of a transistor failure in the capacitive-sensing circuit of a fabricated chip [9] is shown in Fig. 4(b). Traditional fault models, such as stuck-at faults and bridging faults, can be utilized in these cases.

3) Dielectric breakdown. High voltages applied to electrodes can cause dielectric breakdown, which leads to the direct

exposure of a droplet to high voltage and results in droplet electrolysis. Dielectric breakdown is illustrated in Fig. 4(c) for a fabricated chip. In this case, droplet electrolysis occurs, and the droplet cannot be controlled.

4) Short-circuited microelectrodes. A short between two adjacent electrodes leads to a “larger” macro-electrode and the two electrodes cannot be controlled independently. Moreover, once a droplet resides on this macro-electrode, it is no longer possible to create the interfacial surface tension along the droplet transportation path.

III. FAULT-RECOVERY DESIGN

In this section, we introduce the fault-recovery design for each MC. The main idea is as follows. Because the design of each MC is identical, when a fault occurs in an MC, it can use the hardware resource in a neighboring MC to operate correctly. We present four fault-recovery designs for the DFFs as well as for the activation and sensing circuits.

A. Fault-Recovery Design for MC Activation

Before presenting the fault-recovery design, we first introduce some definitions. In our design, each faulty MC will use the hardware resources of neighboring MCs. We define these neighboring MCs as the *backup MCs*. Note that in [15], we have to select downstream MCs as the backup MCs, and the number of backup MCs is limited by the size of the added multiplexer. Suppose the number of the added multiplexers is N , then the number of backup MCs is equal to N . If the value of N increases, the fault-recovery capability is higher, but the area overhead introduced by the added multiplexers is high, which has a higher probability to have faults and adversely affect the fault-recovery operation. The optimum value of N is reported to be “1” in [15] through simulations.

The fault-recovery design for MC activation is shown in Fig. 5(a). For simplicity, we only show the designs for three MCs. For MC activation, a MUX is added to the upper-right corner of each MC (namely MC_i), such that the bottom plate in MC_i is connected to either the output of the Act_i or the output of the MUX from $MC_{(i+1)\%N_{sc}}$, where N_{sc} is the length of the scan chain, and $\%$ refers to the modulo operation. The connection to the bottom plate in MC_i is determined by the control signal, namely ACT_i .

Using this design, we are able to ensure correct MC activation even if the activation circuit or the DFF is faulty. In Fig. 5(a), suppose Act_1 is faulty (i.e., cannot be grounded), we can configure $ACT_1 = 1$ and $ACT_2 = 0$, such that the output of Act_2 is connected to the bottom plate of MC_1 . In this case, if Act_2 is fault-free, we can ensure correct MC activation for MC_1 using Act_2 . Likewise, in Fig. 5(a), if DFF_1 is faulty (e.g., it cannot provide an appropriate MC-activation value), we can again configure $ACT_1 = 1$ and $ACT_2 = 0$, such that the output of Act_2 is connected to the bottom plate of MC_1 . In this case, if both Act_2 and DFF_2 are fault-free, we can ensure correct MC activation for MC_1 using Act_2 and DFF_2 .

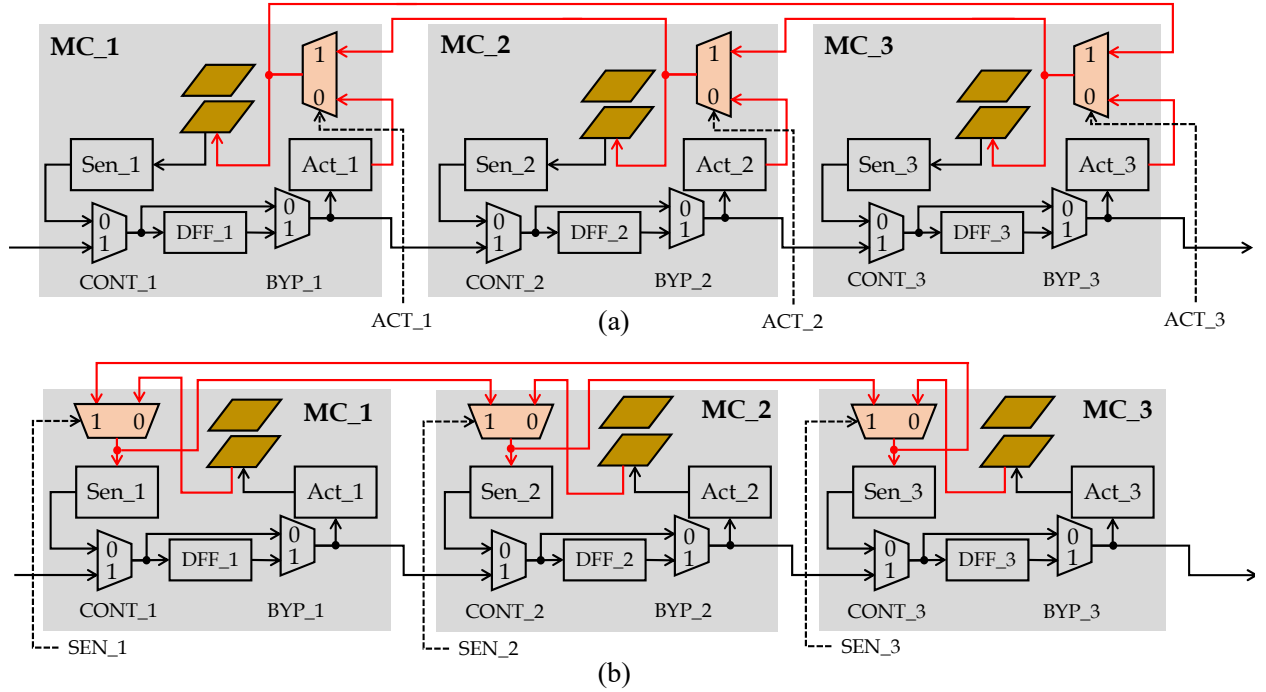


Fig. 5: Fault-recovery designs for (a) MC activation, and (b) MC sensing.

B. Fault-Recovery Design for MC Sensing

The fault-recovery designs for MC sensing is shown in Fig. 5(b). For MC sensing, a MUX is added to the upper-left corner of each MC (namely MC_i), such that the sensing circuit in MC_i can measure the capacitance between the upper and bottom plates of either MC_i or $MC_{(i-1)\%N_{sc}}$, where N_{sc} is the length of the scan chain, and $\%$ refers to the modulo operation. The connection to the sensing circuit in MC_i is determined by the control signal, namely SEN_i .

Using this design, we are able to ensure correct MC sensing even if the sensing circuit or the DFF is faulty. In Fig. 5(b), suppose that Sen_1 is faulty (i.e., cannot measure the capacitance), we can configure $SEN_1 = 0$ and $SEN_2 = 1$, such that the bottom plate of MC_1 is connected to Sen_2 . In this case, if Sen_2 is fault-free, we can ensure correct MC sensing for MC_1 using Sen_2 . Likewise, in Fig. 5(a), if DFF_1 is faulty (i.e., cannot store the MC-sensed value), we can again configure $SEN_1 = 0$ and $SEN_2 = 1$, such that the bottom plate of MC_1 is connected to Sen_2 . In this case, if both Sen_2 and DFF_2 are fault-free, we can ensure correct MC sensing for MC_1 using Sen_2 and DFF_2 .

C. Scan-Chain Testing and Diagnosis

In an MC design, the MUX controlled by $CONT_i$ is used to select between shift data and the MC-sensed value for DFF_i while the MUX controlled by BYP_i is used to create a bypass path, such that even if a DFF is faulty, data shifting is still possible by skipping the faulty DFF.

In the MC-activation operation, “0/1” MC-activation values are serially shifted in DFFs through the scan chain to control the activation status of each MC. In the MC sensing operation, “0/1” MC-sensed values generated by the sensing circuits are stored in DFFs, and then they are serially shifted out through

the scan chain. In addition, when we recover an MC from a fault, the scan chain is also utilized. Therefore, the functionality of the scan chain is of vital importance for both normal MC-activation/sensing operations and fault-recovery operations.

The work in [11] can detect if the scan chain is faulty, but it cannot identify the type of the fault (e.g., faulty DFF, faulty MUX or open fault between MCs). The work in [14] proposed a method to locate faulty DFFs in a scan chain. However, it assumed that the MUXes controlled by $CONT_i$ and BYP_i are fault-free. This assumption is not practical because these two types of MUXes can also have faults and these faults can also lead to failure of the scan chain. In other words, if scan-chain failure is caused only by faulty DFF(s), these faulty DFF(s) can be located and recovered using the method proposed in [14]. Otherwise, the scan chain is not usable and will be blocked.

In order to identify the type of faults in the scan chain, we propose a fault-diagnosis method that includes two stages:

Stage 1: Fault detection.

1) Configure $CONT_i = 1$ and $BYP_i = 0$ ($1 \leq i \leq N_{sc}$), where N_{sc} is the length of the scan chain, such that all DFFs are bypassed and the input to the scan chain is directly connected to the output of the scan chain.

2) Test pattern “0101” (**pattern 1**) of length 4 is applied to the scan chain to detect any fault in the MUXes controlled by BYP_i and also detect open faults between MCs.

3) Configure $CONT_i = 1$ and $BYP_i = 1$ ($1 \leq i \leq N_{sc}$), such that all DFFs are serially connected and no DFF is bypassed.

4) Test pattern “1111...” (**pattern 2**) of length N_{sc} is applied to the scan chain to detect all stuck-at-0 faults in DFFs.

5) Test pattern “0000...” (**pattern 3**) of length N_{sc} is applied to the scan chain to detect all stuck-at-1 faults in DFFs.

6) Test pattern “00110011...” (**pattern 4**) of length $N_{sc} + 4$ is applied to the scan chain to detect delay faults in the scan

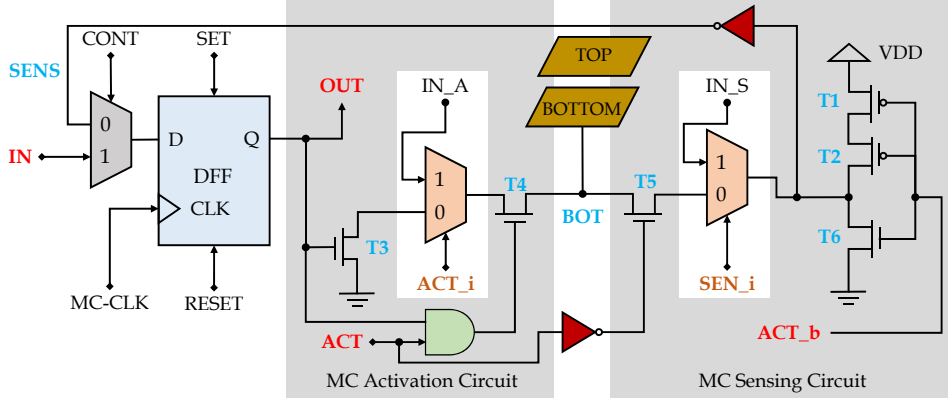


Fig. 6: New hardware design for the microelectrode-cell (MC).

chain.

Stage 2: Fault diagnosis.

1) If the scan chain does not pass *pattern 1* test, it means that the scan-chain failure is caused by either faulty MUXes controlled by BYP_i or open faults between MCs. In this case, the scan chain is regarded to be not usable, and no further diagnosis is carried out.

2) If the scan chain passes *pattern 1* test, and it does not pass one of *pattern 2* test, *pattern 3* test or *pattern 4* test, we first set parameter $j = 1$. Next, we configure $CONT_i = 1, \forall i$, $BYP_j = 0$, and $BYP_i = 0, \forall i \neq j$. Using this configuration, only DFF_j is selected (others are bypassed).

3) Using test patterns “1111”, “0000” and “0101”, we are able to detect stuck-at-0, stuck-at-1 and delay fault for DFF_j .

4) Set parameter $j = j + 1$ and repeat Step 2) and Step 3) of Stage 2 until parameter $j = N_{sc}$, which means all DFFs are individually tested.

Using the above method, we are able to distinguish between two situations: (1) the scan-chain failure is caused by faulty MUXes or open faults (if we fail the test in Step 2) of Stage 1; (2) the scan-chain failure is only caused by faulty DFFs (if we passed the test in Step 2) of Stage 1. In addition, using the steps in Stage 2, we are able to locate faulty DFFs and identify the fault type.

D. Comparison Between Test Methods

In the fault-recovery design reported in [15], the number of backup MCs is an important parameter. Suppose the number of backup MCs is equal to 1, and MC_2 is the backup MC for MC_1 . If DFF_1 in MC_1 is faulty, we can recover from this fault using only DFF_2 in MC_2 . However, if DFF_2 in MC_2 is also faulty, it is impossible to recover from this fault. An easy way to solve this problem is to increase the number of backup MCs, because more backup MCs can be used for fault-recovery. Returning to the previous example, suppose that the number of backup MCs is increased from 1 to 2, and MC_2 and MC_3 are the backup MCs for MC_1 . Even if DFF_2 in MC_2 is faulty, we can still use DFF_3 in MC_3 as a backup to do fault-recovery.

However, increasing the number of backup MCs leads to more control inputs for the recovery MUXes. It is reported in [15] that, if the number of backup MCs is equal to N , a total

of $M \times (4 \times \lceil \log_2(N + 1) \rceil + 1)$ control signals are needed for M MCs in the scan chain. For example, suppose the number of backup MCs is equal to 2. For a fabricated MEDA biochip with 1800 MCs, we need to generate $1800 \times 10 = 18000$ control signals for the multiplexers.

In addition, the increase in the number of backup MCs also lead to larger area overhead. According to [15], the recovery MUXes can be placed under each MC if and only if the number of backup MCs is equal to 1. However, if the number of backup MCs is larger than 1, there is not enough space under each MC to place all the recovery MUXes, and some of the recovery MUXes have to be placed on the boundary of the MEDA biochip.

For the fault-recovery design in this section, we did not discuss the number of backup MCs because each MC in the scan chain can be configured as a backup MC. For example, in Fig. 5(a), if both Act_1 and Act_2 are faulty, we can configure $ACT_1 = 1$, $ACT_2 = 1$, and $ACT_3 = 1$, such that the output of Act_3 is connected to the bottom plate of MC_1 . In this case, we can ensure correct MC activation for MC_1 using Act_3 .

Because the number of recovery MUXes added to each MC is 2, and each MUX has only two inputs, the number of control signals for a scan chain with M MCs is $3M$. For a fabricated MEDA biochip with 1800 MCs, only $1800 \times 3 = 5400$ control signals are needed, which is considerably few than the solution reported in [15]. Note that the control signals for recovery MUXes are not directly controlled by chip pins, as they are generated by configuring the DFFs in sub-scan chains (SSCs) in the IJTAG network; see Section V. With fewer control signals, the configuration time will be shorter. In addition, the proposed design only needs to add two MUXes to each MC, and there is sufficient area under each MC to place these two MUXes removing area overhead concerns.

IV. ROBUST MICROELECTRODE-CELL DESIGN

In this section, we propose an advanced microelectrode-cell (MC) design such that the MC-activation circuit and the MC-sensing circuit are physically separated (i.e., no hardware sharing), and the area used by the MC design is reduced.

TABLE II: Transistor usage for different MC designs.

Original MC Design	Transistors	T1	T2	T3	T4		
	MC Activation	X	X	X	X		
MC Sensing	X	X	X	X			
Proposed MC Design	Transistors	T1	T2	T3	T4	T5	T6
	MC Activation			X	X		
	MC Sensing	X	X			X	X

Note: "X" indicates transistor in use.

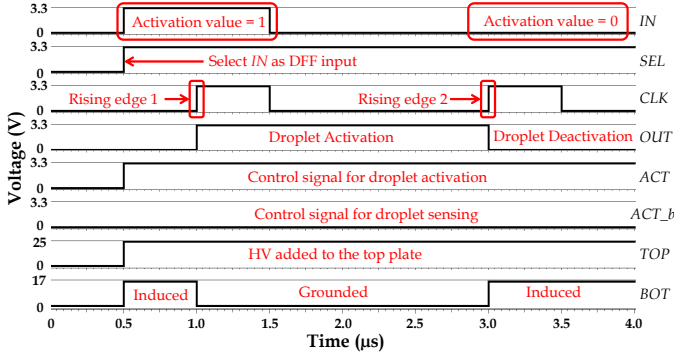


Fig. 7: Simulation results for the MC-activation operation.

A. Motivation

The detailed working principle for the original MC design was described in Section II.A. Recall that both MC activation and MC sensing operations use transistors T1, T2, T3 and T4, which means that the MC activation circuit as well as the MC sensing circuit share hardware. However, this design is not robust because if the MC activation circuit is faulty, the MC sensing circuit is faulty as well, and vice versa.

In order to address this problem, we construct an MC activation circuit and an MC sensing circuit using two separate sets of components. In this case, if one of the circuits (i.e., either an MC activation circuit or an MC sensing circuit) is faulty, it would not affect the other one. Based on this idea, we develop a new MC design such that the MC activation/sensing circuits are separated; see Fig. 6. The comparison of transistor usage for the original design with the proposed design is shown in Table II.

B. Proposed Design and Working Principle

The proposed MC design still includes four parts: a micro-electrode (a top plate + a bottom plate), a DFF, an activation circuit, and a sensing circuit; see Fig. 6. In this design, we also include fault-recovery MUXes for MC-activation and MC-sensing.

We evaluated the MC design of Fig. 6 using HSPICE and a 350 nm library from a foundry. The simulation results (see Fig. 7 and Fig. 8) as well as the new working principle are described as follows:

1) MC activation. In this mode of operation, the control circuit sets $CONT = 1$, $ACT = 1$, $IN = 1$, and a 25 V voltage is applied to the top plate [9]. If a rising edge of MC-CLK is applied to the DFF, pin Q outputs logic "1", and transistors T3 and T4 are switched on while transistor T5 is switched off (i.e.,

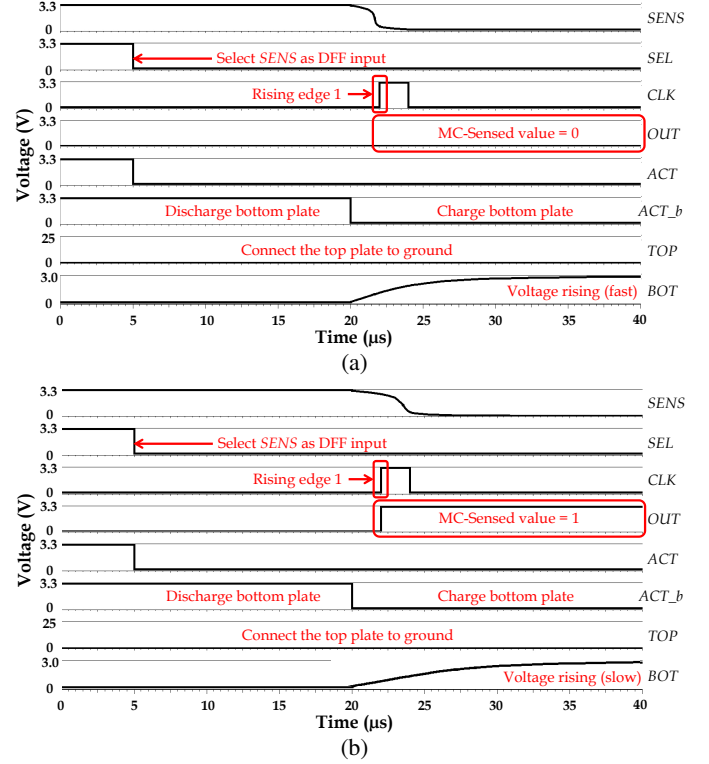


Fig. 8: Simulation results for the MC-sensing operation: (a) with droplet and (b) without droplet.

the MC sensing circuit is disconnected from the bottom plate). In this case, the bottom plate is directly connected to the ground (0 V). The potential difference will generate a force that can drag a droplet towards itself (between rising edge 1 and rising edge 2 of MC-CLK in Fig. 7).

2) MC de-activation. In this mode of operation, the control circuit set $CONT = 1$, $ACT = 1$, $IN = 0$, and a 25 V voltage is applied to the top plate [9]. If a rising edge of MC-CLK is applied to the DFF, pin Q outputs logic "0", and transistors T3 and T4 are switched off. In this case, when a high voltage of 25 V is still applied to the top plate, the bottom plate is floating and an induced voltage of 17 V is generated [16]. Because the potential difference is smaller than the threshold, no force is generated (after rising edge 2 of MC-CLK in Fig. 7).

3) MC sensing. In this mode of operation, the control circuit first set $CONT = 0$, $ACT = 0$, $ACT_b = 1$, and the top plate is connected to the ground. In this case, transistors T5 and T6 are switched on while transistors T1, T2 and T4 are switched off (i.e., the MC activation circuit is disconnected from the bottom plate), which discharges the bottom plate to a voltage of 0 V. Next, the control circuit set $ACT_b = 0$, and transistors T1 and T2 are switched on while transistor T6 is switched off. As a result, the bottom plate is now connected to VDD (3.3 V), and the voltage of net BOT is rising because of capacitor charging. By applying a rising edge of MC-CLK at a preset time, a value of "0" or "1" will be store in the DFF. If a droplet is present between the top plate and the bottom plate, the charging rate is slower, and a value of "1" is expected, which indicates that a droplet is present; otherwise, no droplet is present. Note that the control signal for the fault-tolerance MUX in Fig. 5(b) is

named “SEN”. On the other hand, signal “SENS” in Fig. 8 is the signal that comes out from the sensing circuit. If there is a droplet, the switching time of signal “SENS” will be small; Otherwise, the switching time of signal “SENS” will be high.

The major difference between the original design and the proposed design is that the MC activation circuit and the MC sensing circuit are now independent (i.e., there is no hardware sharing). Second, we use fewer logic gates to implement the same control logic to switch the transistors. The original design uses two MUXes (on the right side of the DFF), one NOR gate, one inverter and one NAND gate to implement the control logic. In the proposed design, we need only one AND gate and one inverter for this purpose.

The major advantage that comes with this change is that the area of the MC is considerably reduced. According to Table III in Section IV, the area of the original MC design without recovery MUXes is $1632 \mu\text{m}^2$ while the area of the proposed MC design without recovery MUXes (i.e., the pink MUXes in Fig. 6) is only $1046 \mu\text{m}^2$ (e.g., a 35% reduction).

C. Self Test and Recoverability Test for MC

A structural test method for an MC is reported in [11], but this method is designed for the original MC design with no recovery MUXes. However, if recovery MUXes are added to the circuit, the problem is considerably different because faults can also occur in these recovery MUXes. For example, in Fig. 5(a), if Act_1 is faulty, we can use Act_2 or Act_3 to carry out MC-activation operation in MC_1. However, if the recovery MUXes in MC_1 is also faulty, then fault recovery is not possible in this case. Therefore, we present a test method to test the *functionality* as well as the *recoverability* for each MC.

Stage 1: MC-sensing self test.

1) Configure $\text{CONT}_i = 0$ and $\text{SEN}_i = 0$ ($1 \leq i \leq N_{sc}$), where N_{sc} is the length of the scan chain, in each MC, the bottom plate is connected to the sensing circuit, and the output of the sensing circuit is connected to the DFF.

2) Use the sensing circuit to discharge the bottom plate to 0 V, and then charge the bottom plate to 3.3 V. A rising edge of MC-CLK is applied to the DFF before the “preset” time such that “1”s are latched in the DFFs.

3) Configure $\text{CONT}_i = 1$ and $\text{BYP}_i = 1$ ($1 \leq i \leq N_{sc}$), then shift out the MC-sensed pattern, and check if the output bitstream is all “1”s. If a certain bit is not equal to “1”, the corresponding sensing circuit is faulty (*sensing check 1*).

4) Configure $\text{CONT}_i = 0$ and $\text{SEN}_i = 0$ ($1 \leq i \leq N_{sc}$), where N_{sc} is the length of the scan chain, in each MC, the bottom plate is connected to the sensing circuit, and the output of the sensing circuit is connected to the DFF.

5) Use the sensing circuit to discharge the bottom plate to 0 V, and then charge the bottom plate to 3.3 V. A rising edge of MC-CLK is applied to the DFF after the “preset” time such that “0”s are latched in the DFFs.

6) Configure $\text{CONT}_i = 1$ and $\text{BYP}_i = 1$ ($1 \leq i \leq N_{sc}$), then shift out the MC-sensed pattern, and check if the output bitstream is all “0”s. If a certain bit is not equal to “0”, the corresponding sensing circuit is faulty (*sensing check 2*).

7) The MCs that fails either *sensing check 1* or *sensing check 2*, and the MCs with a faulty DFF are collectively referred to as *sensing-faulty MCs*, and these MCs need fault-recovery.

Stage 2: MC-sensing recoverability test.

1) Set parameter $j = 1$, and select a *sensing-faulty MC*, namely MC_k .

2) Configure $\text{SEN}_k = 0$ and $\text{SEN}_{(k+i)\%N_{sc}} = 1$ ($1 \leq i \leq j$), where $\%$ refers to the modulo operation, such that the bottom plate of MC_k is connect to $\text{Sen}_{(k+j)\%N_{sc}}$.

3) If $\text{MC}_{(k+j)\%N_{sc}}$ is also a *sensing-faulty MC*, skip to Step 8).

4) Configure $\text{CONT}_{(k+j)\%N_{sc}} = 0$ and use $\text{Sen}_{(k+j)\%N_{sc}}$ to discharge the bottom plate in MC_k to 0 V, and then charge the bottom plate. A rising edge of MC-CLK is applied before the “preset” time such that an MC-sensed value of “1” will be stored in $\text{DFF}_{(k+j)\%N_{sc}}$.

5) Configure $\text{CONT}_i = 1$ ($1 \leq i \leq N_{sc}$), $\text{BYP}_{(k+j)\%N_{sc}} = 1$ and $\text{BYP}_i = 0$ ($i \neq (k+j)\%N_{sc}$), such that only $\text{DFF}_{(k+j)\%N_{sc}}$ is selected. Next, shift out the MC-sensed value and check if it is equal to “1” (*sensing check 3*).

6) Configure $\text{CONT}_{(k+j)\%N_{sc}} = 0$ and use $\text{Sen}_{(k+j)\%N_{sc}}$ to discharge the bottom plate in MC_k to 0 V, and then charge the bottom plate. A rising edge of MC-CLK is applied after the “preset” time such that an MC-sensed value of “0” will be stored in $\text{DFF}_{(k+j)\%N_{sc}}$.

7) Configure $\text{CONT}_i = 1$ ($1 \leq i \leq N_{sc}$), $\text{BYP}_{(k+j)\%N_{sc}} = 1$ and $\text{BYP}_i = 0$ ($i \neq (k+j)\%N_{sc}$), such that only $\text{DFF}_{(k+j)\%N_{sc}}$ is selected. Next, shift out the MC-sensed value and check if it is equal to “0” (*sensing check 4*).

8) If both *sensing check 3* and *sensing check 4* pass, $\text{MC}_{(k+j)\%N_{sc}}$ can be used to recover the sensing circuit of MC_k , and we go straight forward to Stage 3. Otherwise, it is likely that a fault is located in the fault-recovery circuit, and we need to find another MC as backup. Set parameter $j = j + 1$, and repeat Step 2) to Step 7) until we pass both *sensing check 3* and *sensing check 4* or $j = N_{sc}$.

9) If $j = N_{sc}$ and no backup MC is found, the fault in Sen_k is not recoverable.

Stage 3: MC-activation self test.

1) Shift in pattern “00...0” for all DFFs, configure $\text{BYP}_i = 1$ ($1 \leq i \leq N_{sc}$), such that the activation circuit in each MC drives the bottom plate to 0 V, and the output of the sensing circuit is “1”.

2) Configure $\text{CONT}_i = 0$ ($1 \leq i \leq N_{sc}$) and apply a rising edge of MC-CLK to DFFs, then shift out the MC-sensed values and check if the output bitstream is all “1”s. If a certain bit is not equal to “0”, the corresponding activation circuit is deemed to be faulty (*activation check 1*).

3) Shift in pattern “11...1” for all DFFs, configure $\text{BYP}_i = 1$ ($1 \leq i \leq N_{sc}$), such that the activation circuit in each MC does not drive the bottom plate (i.e., floating). Then, we apply a 5 V to the top plate. In this case, a 3.3 V is induced in the bottom plate, and the output of the sensing circuit should be 0.

4) Configure $\text{CONT}_i = 0$ ($1 \leq i \leq N_{sc}$), and apply a rising edge of MC-CLK to DFFs, then shift out the MC-sensed values and check if the output bitstream is all “0”s. If a certain bit is not equal to “1”, the corresponding activation circuit is deemed to be faulty (*activation check 2*).

5) The MCs that fails either *activation check 1* or *activation check 2*, and the MCs with a faulty DFF are collectively referred to as *activation-faulty MCs*, and these MCs need fault-recovery.

Stage 4: MC-activation recoverability test.

1) Set parameter $j = 1$, and select a *activation-faulty MC*, namely MC_k .

2) Configure $ACT_k = 0$ and $ACT_{(k+i)\%N_{sc}} = 1$ ($1 \leq i \leq j$), where $\%$ refers to the modulo operation, such that the bottom plate of MC_k is connect to the $Act_{(k+j)\%N_{sc}}$.

3) If $MC_{(k+j)\%N_{sc}}$ is also a *activation-faulty MC*, skip to Step 8).

4) Configure $CONT_{(k+j)\%N_{sc}} = 0$ and use $Act_{(k+j)\%N_{sc}}$, such that the bottom plate in MC_k is connected to ground (0 V). A rising edge of MC-CLK is applied such that the MC-sensed value of “1” will be stored in $DFF_{(k+j)\%N_{sc}}$.

5) Configure $CONT_i = 1$ ($1 \leq i \leq N_{sc}$), $BYP_{(k+j)\%N_{sc}} = 1$ and $BYP_i = 0$ ($i \neq (k+j)\%N_{sc}$), such that only $DFF_{(k+j)\%N_{sc}}$ is selected. Next, shift out the stored MC-activation value and check if it is equal to “1” (*activation check 3*).

6) Configure $CONT_{(k+j)\%N_{sc}} = 0$ and use $Act_{(k+j)\%N_{sc}}$, such that the bottom plate in MC_k is in floating mode. A rising edge of MC-CLK is applied such that the MC-sensed value of “0” will be stored in $DFF_{(k+j)\%N_{sc}}$.

7) Configure $CONT_i = 1$ ($1 \leq i \leq N_{sc}$), $BYP_{(k+j)\%N_{sc}} = 1$ and $BYP_i = 0$ ($i \neq (k+j)\%N_{sc}$), such that only $DFF_{(k+j)\%N_{sc}}$ is selected. Next, shift out the stored MC-activation value and check if it is equal to “0” (*activation check 4*).

8) If we pass both *activation check 3* and *activation check 4*, $MC_{(k+j)\%N_{sc}}$ can be used to recover the activation circuit of MC_k , and we go straight forward to Stage 3. Otherwise, it is likely that a fault is located in the fault-recovery circuit, and we need to find another MC as backup. Set parameter $j = j+1$, and repeat Step 2) to Step 7) until we pass both *sensing check 3* and *sensing check 4* or $j = N_{sc}$.

9) If $j = N_{sc}$ and no backup MC is found, the fault in Act_k is not recoverable.

Flowcharts and working examples are included in the appendix.

V. OVERALL FAULT-RECOVERY DESIGN

In the previous section, we introduced fault-recovery design for each MC. When a fault occurs in a certain MC, it can use the hardware resources in a backup MC to operate correctly. However, due to hardware sharing, an MC cannot be used for normal operation and fault recovery at the same time. Therefore, in this section, we present a fault-recovery control flow, which determines a schedule for faulty and healthy MCs so that no hardware conflict occurs.

A. IJTAG Network for Fault Recovery

Fig. 9 shows the IJTAG network that we are proposing for the fault-recovery design. It primarily includes three parts:

1) **Sub-Scan Chain (SSC).** In the original design of MEDA, all the MCs are sequentially connected together to form a long scan chain. However, this design is not robust, because if any

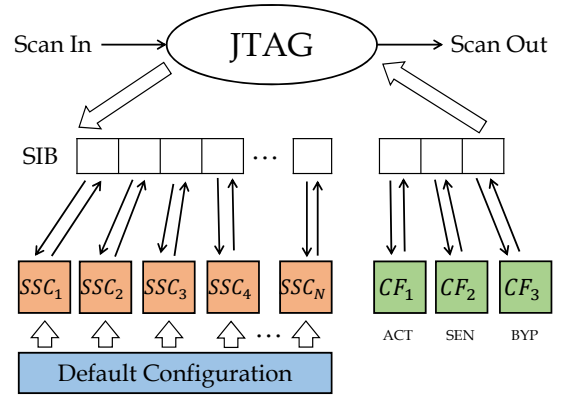


Fig. 9: Illustration of the IJTAG network used for fault recovery.

connection between two MCs is open, data shift is not possible in this case. Therefore, in the IJTAG network, we first divide the scan chain into multiple equal-length SSCs, and then assign each to a SIB register. In this case, even if a connection between two MCs is open, we just need to bypass the non-functional SSC instead of discarding the biochip. As shown in Fig. 9, the red blocks indicate the SSCs, and the subscript represent the index of each SSC. If we have N SSCs, the subscripts will range from 1 to N .

2) **Configuration Registers (CF).** In addition to a more robust scan chain design, the IJTAG network also provides us with access flexibility for SSCs (i.e., each of them can be accessed individually). Rather than recovering from all the faults in the scan chain at the same time, we can do it in a time-multiplexed manner (i.e., recover from the faults in one SSC at a time). Because fault recovery is performed in different SSCs at different times, the SSCs can share the control signals for the multiplexers without any conflict. Therefore, instead of $3M$ control signals in the original design, we now need to generate $3M^*$ control signals, where M is the number of MC in MEDA and M^* is the number of MCs in an SSC.

As shown in Fig. 9, CF_1 to CF_3 (in green color) are the configuration registers. They generate the control signals of ACT, SEN and BYP for a SSC, respectively. Each of these configuration registers is connected to a SIB register; therefore, we can easily change the control signals for multiplexers by shifting the configuration patterns into these configuration registers.

3) **Default Configuration.** In the fault-recovery design, each MC can use the hardware resources of a backup MC. However, if an MC is fault-free (healthy), it will not use any of the backup MCs to operate correctly. In this case, we define the control signals (i.e., ACT, SEN and BYP) set up for a healthy MC as the *default configuration*. In the IJTAG network, the default values of ACT, SEN and BYP are provided to the corresponding multiplexers in the design. For a single SSC, we can choose to configure the multiplexers using either the *default configuration* or a *customized configuration* from the configuration registers (i.e., CF_1 to CF_3).

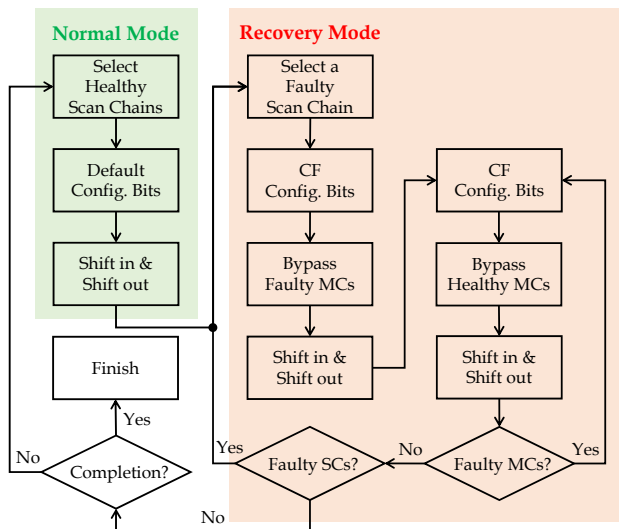


Fig. 10: A fine-grained control flow for fault recovery.

B. Control Flow for Multiplexing

Based on the above IJTAG network, we propose a fine-grained control flow to ensure correct operation of each MC; see Fig. 10. Note that [11], [14] presented methods to detect and locate defects in an MC, therefore, we assume that the *types* and *locations* of defects are known before we use this control flow to recover from faulty MCs. This control flow can be described in terms of two working modes:

1) Normal Mode. In this mode, all healthy SSCs are selected by the IJTAG network (faulty SSCs are not selected), and we configure MCs in healthy SSCs using the default configuration. Next, an activation pattern is shifted into the healthy SSCs, and the sensing results from these SSCs are then shifted out.

2) Recovery Mode. In this mode, fault recovery is performed sequentially for each faulty SSC. First, a faulty scan chain is selected by the IJTAG network. Next, we configure the multiplexers in the selected faulty SSC using the configuration registers such that all the faulty MCs are bypassed, and all the healthy MCs use the default configuration. In this case, we can perform activation-pattern shift-in and sensing-result shift-out for these healthy MCs.

Next, we again configure the multiplexers of the faulty SSC using the configuration registers such that all the healthy MCs are bypassed, and each faulty MC uses the appropriate backup hardware resources. In this case, we can perform activation-pattern shift-in and sensing-result shift-out for these faulty MCs. However, if there are many faults, it is likely that we cannot recover all faulty MCs in only one configuration. Therefore, we need to check whether all the faulty MCs operate correctly using backup MCs. If this is the case, we repeat the same step for the next faulty SSC; otherwise, we continue fault recovery until all the *recoverable* faulty MCs perform their functionality correctly using backup MCs.

Here, a *recoverable* faulty MC is an MC that can recover from faults using backup MCs. However, in some cases, an MC is *not recoverable*. For example, suppose the sensing circuit in MC₁ is faulty, and MC₂ is the only backup MC for MC₁. If MC-activation is performed, fault recovery is not needed. However,

if capacitance-sensing is performed, fault recovery is needed for MC₁. In this case, if the sensing circuit in MC₂ is fault-free, MC₁ is recoverable. Otherwise, MC₁ is not recoverable.

From the above example, we can see that the set of *recoverable* MCs in a faulty SSC can be different for an MC-activation operation and a capacitance-sensing operation. Here, suppose the number of *recoverable* MCs in a faulty SSC are N_{act} and N_{sen} for a MC-activation operation and a capacitance-sensing operation, respectively. Then, the number of multiplexer configurations for this SSC is $N_{act} + N_{sen}$ in the worst case (i.e., fault recovery for one faulty MC in one configuration).

C. Discussion

The clock frequency for the IJTAG network described in this paper is 1 MHz, and the number of clock cycles needed to switch between “Normal Mode” and “Recovery Mode” is less than 1000. Therefore, the time overhead is very small — less than 1 ms. This time scale is negligible compared to the time interval of 1 s between consecutive microfluidic operations.

When we operate a MEDA biochip, the activation pattern will change every one second. Suppose MC-A and MC-B are healthy and MC-C is faulty. At time slot (t, t+1), suppose MC-A, MC-B and MC-C are required to be activated for one second. In this case, we can configure MC-A as the backup for MC-C. By switching between “Normal Mode” and “Recovery Mode” several times (e.g., five times), we can activate MC-A and MC-C in a time-multiplexed manner.

However, in time slot (t, t+1), if MC-B is not required to be activated (i.e., not used), we can configure MC-B as the backup for MC-C. In this case, both MC-A and MC-C can be continuously activated for the whole time slot (t, t+1).

In addition, when we operate a MEDA biochip, the droplet sensing operation is carried out on each MC every one second. Since the sensing operation takes 1 ms, we can simply carry out droplet sensing for healthy MCs, and then do it for faulty MCs. This order of operations does not introduce any hardware-sharing conflict.

The selection of cells for the sub scan-chain (SSC) does not have impact on recovery time. Therefore, a serial X-Y walk as shown in Fig. 2 would suffice. Also note that the recovery-scheme is used online (i.e., while the assay is running), and it does not have any impact on droplet operations, such as droplet routing, droplet splitting and droplet mixing.

D. An Illustrative Example

In order to make the control flow easier to understand, we use an illustrative example; see Fig. 11. We consider a fabricated MEDA biochip with 1800 MCs [11]. Suppose the scan chain is divided into 60 SSCs; then we have 30 MCs in for each SSC. We also assume that a fault occurs in the sensing circuit of MC1 in SC3. The following three steps need to be carried out for the control flow in Fig. 10:

Step 1: All healthy SSCs (except for SSC₃) are selected by the IJTAG network, and all the healthy sub scan chains use the default configuration. Next, activation pattern shift-in and sensing result shift-out are performed.

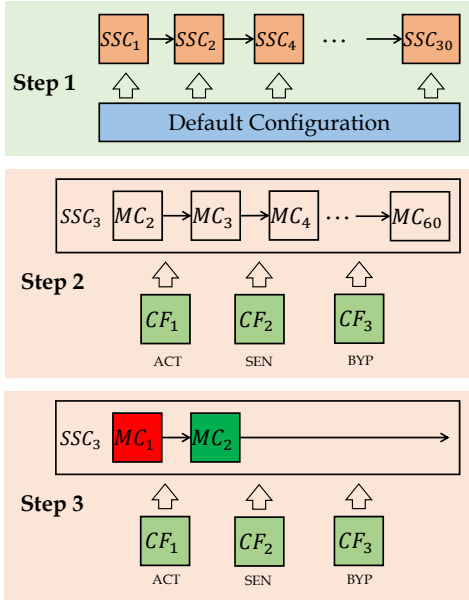


Fig. 11: An illustrative example for the fault-recovery control flow.

TABLE III: The areas of different components in an MC.

Component	Area (μm^2)	Component	Area (μm^2)
T1	15	T2	15
T3	4	T4	40
T5	40	T6	4
NAND/NOR	100	MUX2	200
INV	70	DFF	400
DFF	400	Con. Wire	17.5

Step 2: The faulty SSC_3 is selected by the IJTAG network, and the configuration registers are used such that the faulty MC_1 is bypassed; the rest of the MCs use the default configuration. Next, activation-pattern shift-in and sensing-result shift-out are performed.

Step 3: The faulty SSC_3 is still selected by the IJTAG network, but the configuration registers are used such that only MC_1 (which is faulty) and its backup (MC_2) is selected. Next, MC-activation and capacitance-sensing operations are performed correctly in MC_1 with the help of MC_2 .

Note that for every single activation pattern in a bioassay, the MEDA biochip should go through these three steps until the completion of bioassay execution.

VI. EXPERIMENTAL RESULTS

In this section, we first analyze the area and time overhead of the fault-recovery design. Next, we introduce two evaluation metrics and the simulation setup for the proposed fault-recovery design. Then, we evaluate the benefits of the fault-recovery multiplexer-based design and the IJTAG network. Finally, the optimization of two design parameters is presented.

A. Area and Time Overhead Analysis

According to [11], [22], major concerns for test-circuit insertion and IJTAG network design are area overhead and access-time overhead, respectively. In the case of the IJTAG network,

because we need to configure the SIB registers, additional clock cycles are needed, and the total access time is increased. However, in a MEDA biochip, these two concerns do not arise.

In MEDA, the total area under each MC is $50 \times 50 = 2500 \mu\text{m}^2$ [14]. Using an AMI 0.35 μm PDK [24], we obtain the areas for different standard cells (e.g., INV, MUX and DFF), and the areas corresponding to the components used in an MC are listed in Table III. The dimensions of the standard cells are included in the appendix.

We can see that the areas of the transistors are different. T3/T6 is an NMOS transistor used to connect the bottom plate to the ground, therefore the smallest possible size is used. However, T1 and T2 are used to charge the bottom plate to 3.3V in the capacitance-sensing process. In order to increase the resolution of capacitance sensing, T1 and T2 are intentionally stretched (the length is several times longer than usual) in fabricated MEDA chips to obtain a small capacitor-charging current [25]. Finally, T4/T5 is an NMOS transistor that need to endure a voltage of up to 15 V when the MC is activated [9], therefore it specially designed, and its area is much larger than a minimum-size NMOS.

According to [24], the minimum trace width is 0.7 μm . Suppose the interconnect between two neighboring MCs is 25 μm (i.e., 1/2 of the MC width). The area of the interconnect wire (denoted as Con. Wire) is 17.5 μm^2 . In addition, the area of the BYP/CONT/ACT/SEN MUXes are all 200 μm^2 .

In the proposed fault-recovery solution, each MC needs one BYP MUX, one ACT MUX and one SEN MUX. The area of the new MC design proposed in Section IV is 1646 μm^2 . However, the area under an MC is 2500 μm^2 , which implies that we can fit the new MC design under an MC without any area overhead.

In an IJTAG network, extra clock cycles (a.k.a retargeting time) to configure the SIB registers are needed. However, because the clock signal in MEDA has a frequency of 1 MHz, the time to shift in activation patterns (1800 bits) and shift out sensing results (1800 bits) is only 3.6 ms, and this data-shift process is needed only once per second. Therefore, access-time overhead is also not a concern, and sufficient time is available for fault recovery.

B. Evaluation Metrics

Therefore, rather than focusing on area overhead and access-time overhead, we introduce two metrics that can be used to evaluate the effectiveness of fault-recovery design, namely *fault-recovery rate* (FR) and *normalized data-shift* (DS).

The first evaluation metric FR is defined as follows:

$$FR = \frac{N_{ar} + N_{sr}}{N_{af} + N_{sf}} \quad (1)$$

The major functionalities of an MC include MC activation and capacitance sensing. If a fault occurs either in the activation circuit or in the DFF, we refer to this situation as an MC-activation fault. On the other hand, if a fault occurs either in the sensing circuit or in the DFF, this situation is referred to as a capacitance-sensing fault. Referring back to Equation (1), N_{af} and N_{sf} indicate the number of MCs that have MC-activation fault and capacitance-sensing fault, respectively. In addition,

N_{ar} and N_{sr} represent the number of MCs that can recover from MC-activation fault and capacitance-sensing fault, respectively. The FR metric can be used to evaluate the fault recoverability of the proposed design.

The second evaluation metric DS is defined as follows:

$$DS = \frac{N_s}{3600} \quad (2)$$

where N_s is the number of bits shifted per second in the fault-recovery design, and “3600” represents 3600 bits are shifted in the scan chain per second in the original MEDA design (1800 bits for activation pattern and 1800 bits for capacitance-sensing results). The value of N_s can be approximately calculated as follows:

$$N_s \approx 3600 + (1 + N_{fs}) \times N_s + \sum_{N_{fs}}^i N_{fm}(i) \times (4 \times \log_2(N_b) + 2) \quad (3)$$

where N_{fs} is the number of faulty SSCs, N_s is the total number of SSCs, $N_{fm}(i)$ is the number of faulty MCs that can be recovered in the i th SSC, and N_b is the number of backup MCs for each MC.

The evaluation metric DS is important, because a higher value of DS indicates higher power consumption in MEDA, which is likely to heat up the biochip, and adversely impact the execution of temperature-sensitive bioassays. In addition, a higher value of DS also indicates more frequent use of DFFs, which is likely to reduce hardware reliability and lead to performance degradation.

C. Fault Simulation Setup

In this subsection, we first describe how to inject faults in an MC, and then describe how to determine whether a faulty MC can be recovered.

In order to evaluate the proposed fault-recovery design, we randomly inject multiple faults into the MEDA biochip, then we count the total number of faulty MCs, and also the numbers of faulty MCs that can be recovered from the injected faults. As a result, we can compute the two evaluation metrics (FR and DS) based on Equation (1) and Equation (2). To make the simulation more realistic, faults injected into the MCs target not only the original circuit, but also the multiplexers used for fault recovery.

When a fault is injected into a given MC, the probability that a functional block has this fault is proportional to the area it occupies. As shown in Fig. 1(a), the transistors/gates for the circuit under each MC include transistors T1, T2, T3 and T4, three 2-input MUXes, one DFF, one NAND gate, one NOR gate, and two inverters. For example, if we want to inject a fault into the original circuit, because the area of the DFF and the original circuit are $400 \mu\text{m}^2$ and $1414 \mu\text{m}^2$, the probability that the fault is injected into the DFF is $400/1414 = 0.282$.

When a fault is injected into a functional block in the original design, such as the activation circuit, sensing circuit or DFF, we assume that the functional block fails. However, when a fault is injected into one of the fault-recovery multiplexers, we assume

TABLE IV: The criteria to determine the type of fault for an MC.

Types of Faults	Act. Cir.	Sen. Cir.	DFF	Con. Wire	BYP MUX	ACT MUX	SEN MUX
MC Activation	F		F		F-2	F-2 F-3	
MC Sensing		F	F		F-2		F-2 F-3
Broken Scan Chain				F	F-3		

Note: F indicate a failure in the functional block.

the following three types of faults: **(F-1)** The multiplexer cannot select a component from a backup MC; **(F-2)** The multiplexer cannot select a component from a local MC (the MC where the multiplexer is located); **(F-3)** The multiplexer is faulty, and it cannot select any component. In the simulation, when a fault appears in a fault-recovery multiplexer, we randomly select a fault type from the three of them.

After faults are injected into the MCs in MEDA, we determine whether an MC has an MC-activation fault or a capacitance-sensing fault in order to compute the value of FR and DS. The criteria to determine the type of fault for an MC is listed in Table IV. Note that if any one of the faults listed in the columns occurs, then an MC is deemed to have the corresponding fault. The third row in Table IV (i.e., broken scan-chain fault) is a special case. If this fault occurs, an MC is deemed to have both an MC-activation and a capacitance-sensing faults, because the MC cannot be accessed in this case.

The method to determine whether an MC can be recovered from these two faults is simple. For an MC with an MC-activation fault, if both a fault-free activation circuit and a fault-free DFF can be selected from its own or from its backup MCs, this MC is able to recover from an MC-activation fault. Otherwise, recovery is not possible. A similar procedure can be used for an MC with a capacitance-sensing fault.

D. Benefits of the Proposed Design

In this subsection, we quantify the benefits introduced by the fault-recovery design and the IJTAG network. In order to do that, we compare the following three designs:

D-1: The fault-recovery design presented in [14]. In this design, three redundant DFFs are added to each MC, and it can recover only from faulty DFFs.

D-2: The fault-recovery design proposed in [15] with one backup MC for each MC, but without the IJTAG network.

D-3: The fault-recovery design proposed in [15] with one backup MC for each MC. The complete MEDA scan chain is divided into 60 SSCs, which are managed using an IJTAG network.

D-4: The proposed fault-recovery design using the new MC design. The complete MEDA scan chain is divided into 60 SSCs, which are managed using an IJTAG network.

Two simulations are needed to quantify the benefits — In the first simulation (Sim-1), faults can be freely injected into any functional blocks. In this simulation, we compare the average FRs of D-1 to D-4 to see the benefit introduced by the IJTAG network. In the second simulation (Sim-2), no fault is injected

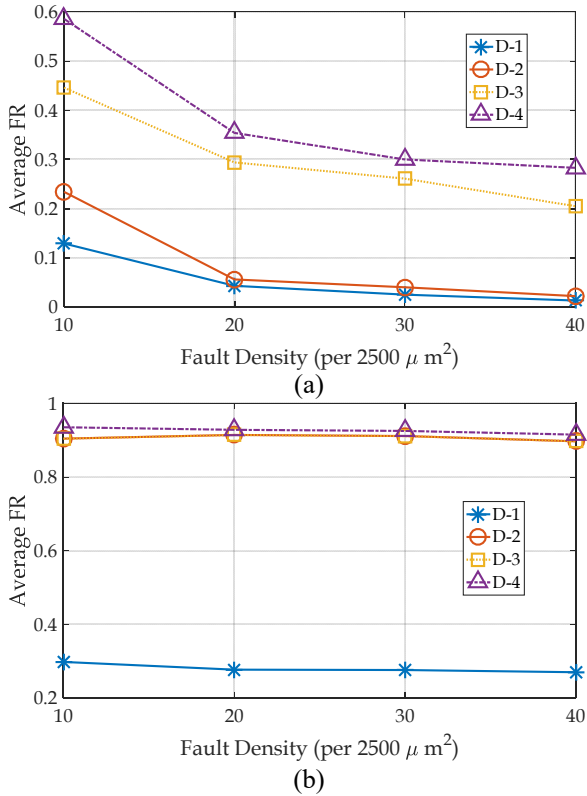


Fig. 12: The average FRs for D-1, D-2, D-3 and D-4 in (a) Sim-1 and (b) Sim-2.

in the connection wire, and no F-3 fault is injected into the BYP MUX, so that broken scan-chain fault will not occur in MEDA. In this simulation, we compare the average FRs of D-1 to D-4 to compare the fault-recovery capability of different fault-recovery designs.

The simulation results are shown in Fig. 12. We can see that, in Sim-1, because of the broken scan-chain fault, the FR values are much lower for D-1 and D-2 because the scan chain is not usable and, fault-recovery cannot be carried out in this case. However, for D-3 and D-4, because of the utilization of IJTAG network, when a broken scan-chain fault occurs, we only need to bypass the faulty SSCs, and can still access the remaining SSCs. Therefore, the FR values for these two methods are much higher than those for D-1 and D-2.

In the second simulation (Sim-2), because no broken scan-chain faults occurs, we can compare the fault-recovery capability of all the four designs. The FR value for D-1 is still very low because this design can only recover from DFF faults, and it cannot handle other types of faults. On the other hand, D-2, D-3 and D-4 all achieves FR values of 0.90, which indicates that the fault-recovery solutions in [15] and in this work are effective. Note that, the FR value for D-4 is slightly higher than that for D-2/D-3 because every MC in the scan chain can be configured as a backup MC in D-4 while only a downstream MC can be configured as a backup MC in D-2/D-3.

E. Design Optimization

In this subsection, we optimize a design parameter in the proposed fault-recovery design, namely the number of SSCs

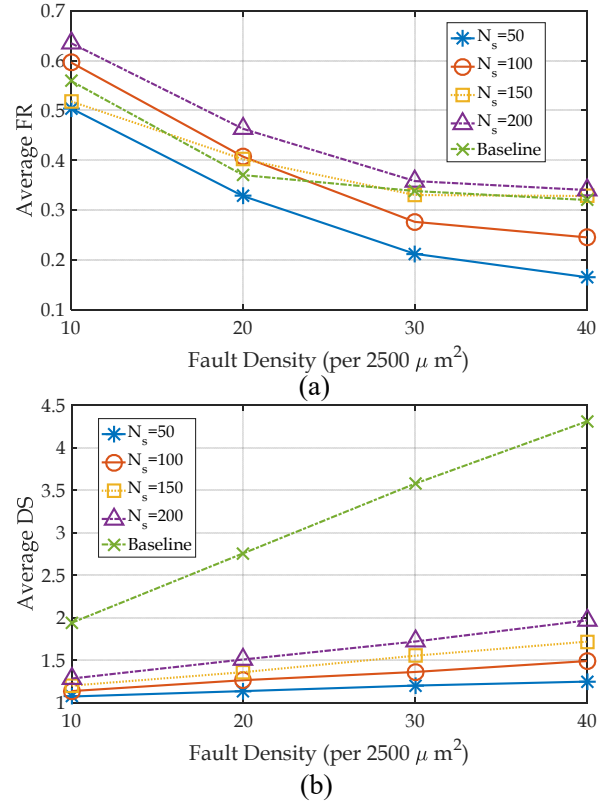


Fig. 13: The FR and DS values for different numbers of SSCs.

(N_s). Note that the method proposed in [15] (with 200 SSCs and the number of backup MCs is 1) is used as the baseline method.

In order to do this, we increase the number of SSCs from 50 to 200 with increments of 50. Then, we obtain the simulation results as shown in Fig. 13. We can see that higher N_s leads to a higher average FR because with a shorter SSC, fewer MCs will be affected by a broken scan-chain fault. On the other hand, we also note a higher value of DS with an increase in N_s . With more SSCs, the length of the SIB register will increase, which indicates a higher retargeting cost (i.e., the length of the bitstream needed to configure the SIB registers and state transition for the TAP controller) for the IJTAG network. However, according to Fig. 13, the DS value is only 2 when the fault density is 40 faults per 2500 μm^2 and $N_s = 200$. This value is much lower than that reported in [15] ($DS = 5$). Therefore, we can regard this DS increment as negligible, and $N_s = 200$ is thus an optimum value for this design.

VII. CONCLUSION

We have presented an efficient fault-recovery solution for emerging MEDA biochips. Since the microelectrode cell (MCs) in a MEDA biochip are identical, we have added two types of MUXes for reconfigurability, such that an MC with faulty components can use the hardware resources in a downstream MC. In addition, we have proposed a robust MC design that makes the MC activation circuit and the MC sensing circuit independent, and this design leads to much lower area overhead than the original design. Finally, we have used the IEEE 1687 (i.e., IJTAG) network to reduce the number of control signals

needed for the MUXes, and to provide flexible sub-scan chain access for the fault-recovery control flow. A comprehensive set of simulation results demonstrate the benefits of the proposed fault-recovery design and the IJTAG network. The proposed design (using the new MC design + fault-recovery MUXes) only consumes 1646 μm^2 area, which is only 65% of the 2500 μm^2 under a microelectrode.

ACKNOWLEDGMENT

The authors thank August Ning from Duke University for carefully proof-reading a draft version of this manuscript.

REFERENCES

- [1] E. Samiei *et al.*, "A review of digital microfluidics as portable platforms for lab-on-a-chip applications," *Lab on a Chip*, vol. 16, no. 13, pp. 2376–2396, 2016.
- [2] K. Choi *et al.*, "Digital microfluidics," *Annual Review of Analytical Chemistry*, vol. 5, pp. 413–440, 2012.
- [3] F. Su *et al.*, "Ensuring the operational health of droplet-based microelectrofluidic biosensor systems," *IEEE Sensors Journal*, vol. 5, no. 4, pp. 763–773, 2005.
- [4] S. O. Oyola *et al.*, "Optimizing illumina next-generation sequencing library preparation for extremely at-biased genomes," *BMC Genomics*, vol. 13, no. 1, p. 1, 2012.
- [5] V. M. Pierce *et al.*, "Comparison of the Genmark diagnostics eSensor respiratory viral panel to real-time pcr for detection of respiratory viruses in children," *Journal of Clinical Microbiology*, vol. 50, no. 11, pp. 3458–3465, 2012.
- [6] P. V. Hopkins *et al.*, "Lysosomal storage disorder screening implementation: findings from the first six months of full population pilot testing in missouri," *The Journal of Pediatrics*, vol. 166, no. 1, pp. 172–177, 2015.
- [7] G. Wang *et al.*, "Digital microfluidic operations on micro-electrode dot array architecture," *IET Nanobiotechnology*, vol. 5, no. 4, pp. 152–160, 2011.
- [8] K. Y.-T. Lai *et al.*, "An intelligent digital microfluidic processor for biomedical detection," *Journal of Signal Processing Systems*, vol. 78, no. 1, pp. 85–93, 2015.
- [9] K. Y.-T. Lai *et al.*, "A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35 μm standard cmos process," in *A-SSCC*, pp. 1–4, 2015.
- [10] Y. Ho *et al.*, "Design of a micro-electrode cell for programmable lab-on-cmos platform," in *ISCAS*, pp. 2871–2874, IEEE, 2016.
- [11] Z. Li *et al.*, "Built-in self-test for micro-electrode-dot-array digital microfluidic biochips," in *ITC*, pp. 1–10, 2016.
- [12] Z. Li *et al.*, "Structural and functional test methods for micro-electrode-dot-array digital microfluidic biochips," *TCAD*, vol. 37, no. 5, pp. 968–981, 2018.
- [13] V. Shukla *et al.*, "Offline error detection in meda-based digital microfluidic biochips using oscillation-based testing methodology," *Journal of Electronic Testing*, vol. 33, no. 5, pp. 621–635, 2017.
- [14] L. Zhang *et al.*, "Built-in self-diagnosis and fault-tolerant daisy-chain design in meda biochips," in *ITC*, pp. 1–10, 2018.
- [15] Z. Zhong *et al.*, "Fault recovery in micro-electrode-dot-array digital microfluidic biochips using an IJTAG network," in *ITC*, 2019.
- [16] G. Wang, *Field-programmable microfluidic test platform for point-of-care diagnostics*. PhD thesis, University of Saskatchewan, 2013.
- [17] Z. Li *et al.*, "High-level synthesis for micro-electrode-dot-array digital microfluidic biochips," in *DAC*, 2016.
- [18] Z. Zhong *et al.*, "Micro-Electrode-Dot-Array digital microfluidic biochips: Technology, design automation, and test techniques," *TBioCAS*, vol. 13, no. 2, pp. 292–313, 2018.
- [19] IEEE Standard Committee *et al.*, "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Standard*.
- [20] IEEE Standard Committee *et al.*, "IEEE standard test access port and boundary scan architecture," *IEEE Standard*.
- [21] R. Cantoro *et al.*, "Automatic generation of stimuli for fault diagnosis in IEEE 1687 networks," in *IOLTS*, pp. 167–172, 2016.
- [22] F. G. Zadegan *et al.*, "Design automation for IEEE P1687," in *DATE*, pp. 1–6, 2011.
- [23] F. G. Zadegan *et al.*, "Test time analysis for IEEE P1687," in *ATS*, pp. 455–460, 2010.
- [24] "AMI 0.35 μm PDK." <https://vlsiarch.ecen.okstate.edu/flow>, 2019.
- [25] C.-Y. Lee. National Chiao-Tung University, Taiwan, Private Correspondence, 2018.



Zhanwei Zhong received the bachelor's degree in electronic science and technology from Sun Yat-sen University, Guangzhou, China in 2013, and the master's degree in electronic engineering from Tsinghua University, Beijing, China, in 2016.

He is currently a Ph.D. student at Duke University, Durham, NC, USA. He was a DFT intern with AMD Inc. in 2017 and with Apple Inc. in 2019. His current research interests include design automation for biochips and IJTAG test standard.



Krishnendu Chakrabarty (F'08) received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively. He is now the John Cocke Distinguished Professor in the Department of Electrical and Computer Engineering and Professor of Computer Science at Duke University. He also serves as Chair of the Department of Electrical and Computer Engineering. Prof. Chakrabarty is a recipient of the National Science Foundation CAREER award, the

Office of Naval Research Young Investigator award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE Transactions on CAD Donald O. Pederson Best Paper Award (2015), the ACM Transactions on Design Automation of Electronic Systems Best Paper Award (2016), and over a dozen best paper awards at major conferences. He is also a recipient of the IEEE Computer Society Technical Achievement Award (2015), the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award (2017), the IEEE Test Technology Technical Council Bob Madge Innovation Award (2018), and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur (2014). He is a Research Ambassador of the University of Bremen (Germany). He was a Hans Fischer Senior Fellow at the Institute for Advanced Study, Technical University of Munich, Germany, during 2016–2019.

Prof. Chakrabarty's current research projects include: testing and DFT of integrated circuits and systems; digital microfluidics, biochips, and cyberphysical systems; data analytics for fault diagnosis, failure prediction, anomaly detection, and hardware security; smart manufacturing. He is a Fellow of ACM, a Fellow of IEEE, a Fellow of AAAS, and a Golden Core Member of the IEEE Computer Society. He holds nine US patents, with several patents pending. He received the 2018 Invitational Fellowship of the Japan Society for the Promotion of Science (JSPS) in the "Nobel Prize level" category. He is a recipient of the 2008 Duke University Graduate School Deans Award for excellence in mentoring, and the 2010 Capers and Marion McDonald Award for Excellence in Mentoring and Advising, Pratt School of Engineering, Duke University. He has served as a Distinguished Visitor of the IEEE Computer Society (2005–2007, 2010–2012), a Distinguished Lecturer of the IEEE Circuits and Systems Society (2006–2007, 2012–2013), and an ACM Distinguished Speaker (2008–2016).

Prof. Chakrabarty served as the Editor-in-Chief of *IEEE Design & Test of Computers* during 2010–2012, *ACM Journal on Emerging Technologies in Computing Systems* during 2010–2015 and *IEEE Transactions on VLSI Systems* during 2015–2018. He is currently an Associate Editor of *IEEE Transactions on Biomedical Circuits and Systems* and *ACM Transactions on Design Automation of Electronic Systems*.

Appendix

A. Flow Charts

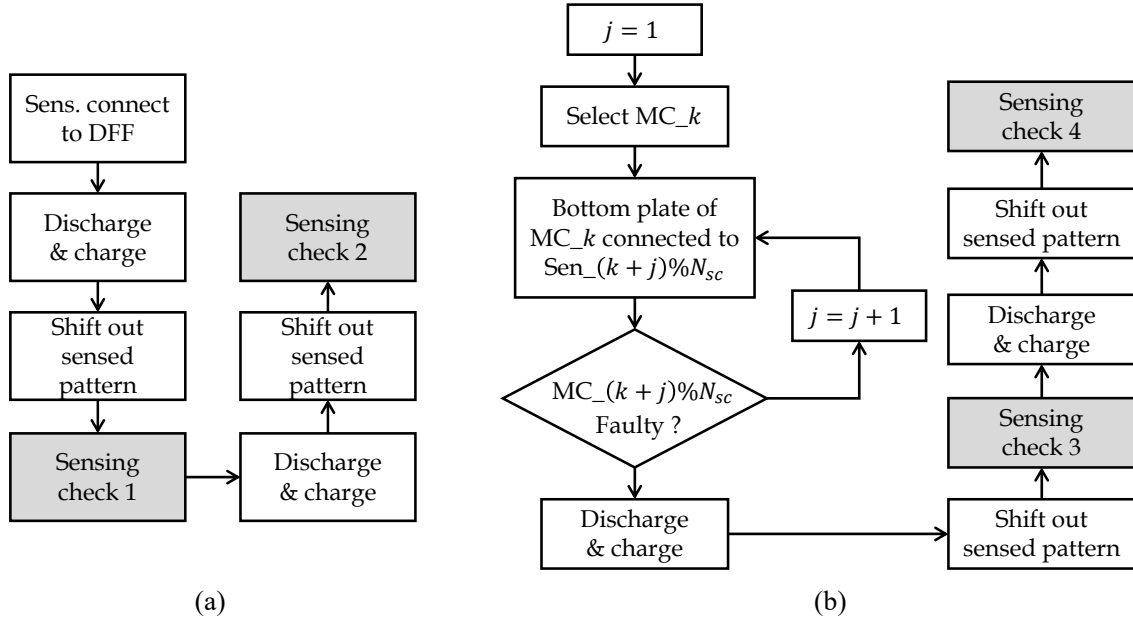


Fig. 1: Flow-charts for (a) MC-sensing self test, and (b) MC-sensing recoverability test.

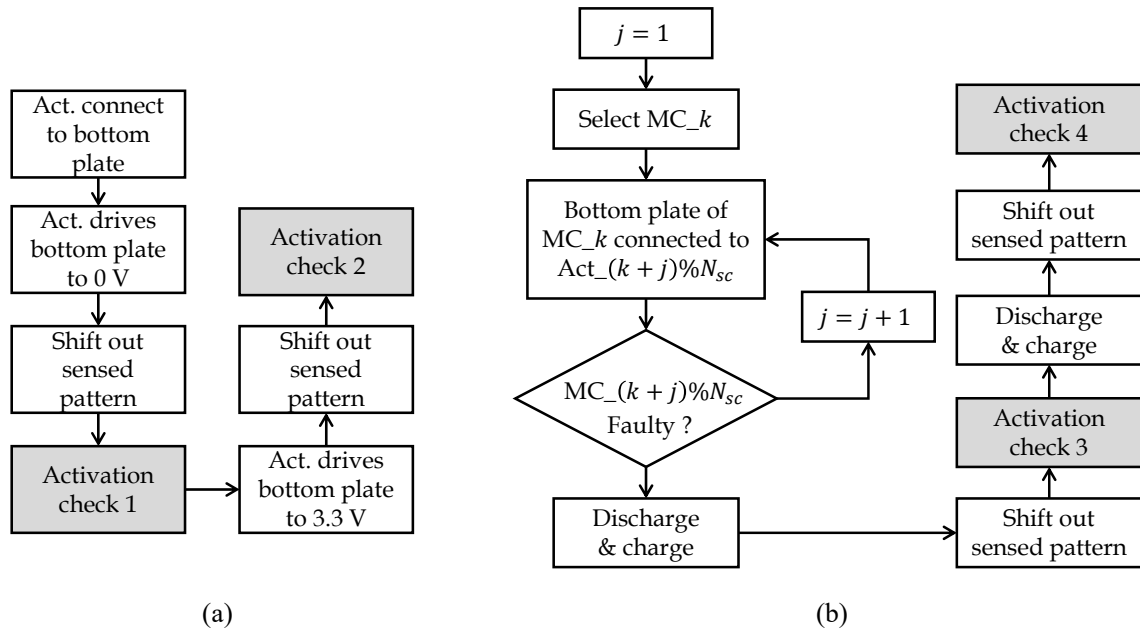


Fig. 2: Flow-charts for (a) MC-activation self test, and (b) MC-activation recoverability test.

B. Working Examples

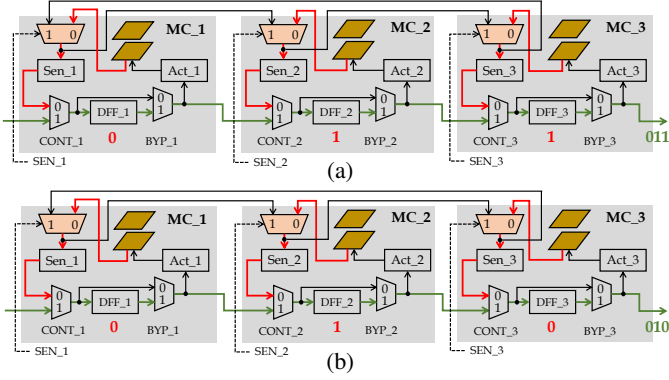


Fig. 1: Illustrations for Stage 1: (a) steps 1), 2) and 3); (b) steps 4), 5) and 6).

Stage 1: MC-sensing self test.

1) Configure $CONT_i = 0$ and $SEN_i = 0$ ($1 \leq i \leq 3$), such that, in each MC, the bottom plate is connected to the sensing circuit, and the output of the sensing circuit is connected to the DFF (indicated by red arrows in Fig. 1(a)).

2) Use the sensing circuit to discharge the bottom plate to 0 V, and then charge the bottom plate to 3.3 V. A rising edge of MC-CLK is applied to the DFF before the “preset” time such that “1”s are latched in the DFFs.

3) Configure $CONT_i = 1$ and $BYP_i = 1$ ($1 \leq i \leq 3$), then shift out the MC-sensed pattern (indicated by green arrows in Fig. 1(a)), and the output bitstream should be “111”. Suppose the output bitstream is “011”, then the sensing circuit in MC_1 is deemed to be faulty (*sensing check 1*).

4) Configure $CONT_i = 0$ and $SEN_i = 0$ ($1 \leq i \leq 3$), such that, in each MC, the bottom plate is connected to the sensing circuit, and the output of the sensing circuit is connected to the DFF (indicated by red arrows in Fig. 1(b)).

5) Use the sensing circuit to charge the bottom plate to 3.3 V, and then discharge the bottom plate. A rising edge of MC-CLK is applied to the DFF after the “preset” time such that an “0”s are latched in the DFFs.

6) Configure $CONT_i = 1$ and $BYP_i = 1$ ($1 \leq i \leq 3$), then shift out the MC-sensed pattern (indicated by green arrows in Fig. 1(b)), and the output bitstream should be “000”. Suppose the output bitstream is “010”, then the sensing circuit in MC_2 is deemed to be faulty (*sensing check 2*).

7) MC_1 and MC_2 are collectively referred to as *sensing-faulty MCs*, and these MCs need fault-recovery.

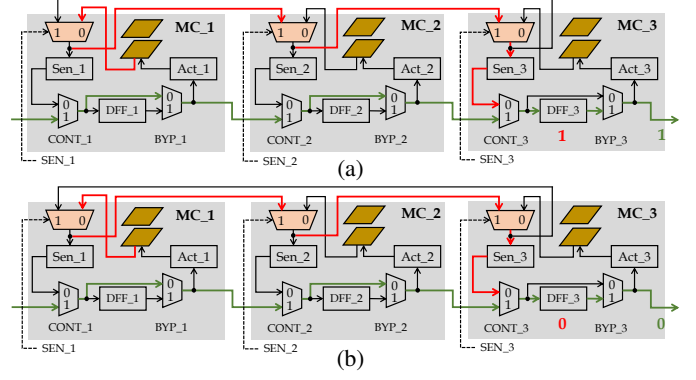


Fig. 2: Illustrations for Stage 2: (a) steps 4) and 5); (b) steps 6) and 7).

Stage 2: MC-sensing recoverability test.

1) Select a *sensing-faulty MC*, for example, MC_1.

2) Configure $SEN_1 = 0$ and $SEN_2 = 1$, such that the bottom plate of MC_1 is connect to Sen_2.

3) Because MC_2 is also a *sensing-faulty MC*, we need to find another MC as a backup. Configure $SEN_1 = 0$, $SEN_2 = 1$, and $SEN_3 = 1$, such that the bottom plate of MC_1 is connect to Sen_3 (indicated by red arrows in Fig. 2(a)).

4) Configure $CONT_3 = 0$ and use Sen_3 to discharge the bottom plate in MC_1 to 0 V, and then charge the bottom plate. A rising edge of MC-CLK is applied before the “preset” time such that an MC-sensed value of “1” will be stored in DFF_3.

5) Configure $CONT_i = 1$ ($1 \leq i \leq 3$), $BYP_1 = 0$, $BYP_2 = 0$, and $BYP_3 = 1$, such that only DFF_3 is selected. Next, we shift out the MC-sensed value and the output bitstream should be “1” (indicated by green arrows in Fig. 2(a)). Suppose the output bitstream is “1”, which passed *sensing check 3*.

6) Configure $CONT_3 = 0$ and use Sen_3 to discharge the bottom plate in MC_1 to 0 V, and then charge the bottom plate. A rising edge of MC-CLK is applied after the “preset” time such that an MC-sensed value of “0” will be stored in DFF_3.

7) Configure $CONT_i = 1$ ($1 \leq i \leq 3$), $BYP_1 = 0$, $BYP_2 = 0$, and $BYP_3 = 1$, such that only DFF_3 is selected. Next, we shift out the MC-sensed value and the output bitstream should be “0” (indicated by green arrows in Fig. 2(b)). Suppose the output bitstream is “0”, which passed *sensing check 4*.

8) Because we passed both *sensing check 3* and *sensing check 4*, MC_3 can be used to recover the sensing circuit in MC_1.

9) A similar process is applied to check if either MC_3 or MC_1 can be used to recover the sensing circuit in MC_2.

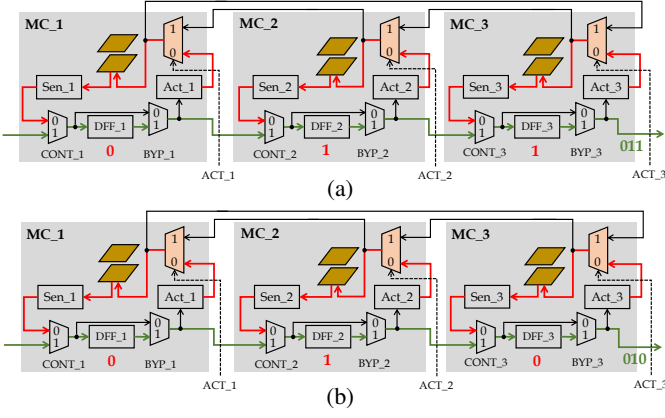


Fig. 3: Illustrations for Stage 3: (a) steps 1) and 2); (b) steps 3) and 4).

Stage 3: MC-activation self test.

1) Shift in pattern “000” for all DFFs, configure $BYP_i = 1$ ($1 \leq i \leq 3$), such that the activation circuit in each MC drives the bottom plate to 0 V, and the output of the sensing circuit is “1” (indicated by red arrows in Fig. 3(a)).

2) Configure $CONT_i = 0$ ($1 \leq i \leq 3$) and apply a rising edge of MC-CLK to DFFs, then shift out the MC-sensed values and check if the output bitstream is “111”. Suppose the output bitstream is “011”, then the activation circuit in MC_1 is deemed to be faulty (*activation check 1*).

3) Shift in pattern “111” for all DFFs, configure $BYP_i = 1$ ($1 \leq i \leq 3$), such that the activation circuit in each MC does not drive the bottom plate (i.e., floating). Then, we apply a 5 V to the top plate. In this case, a 3.3 V is induced in the bottom plate [1], and the output of the sensing circuit should be “0” (indicated by red arrows in Fig. 3(b)).

4) Configure $CONT_i = 0$ ($1 \leq i \leq 3$), and apply a rising edge of MC-CLK to DFFs, then shift out the MC-sensed values and check if the output bitstream is “000”. Suppose the output bitstream is “010”, then the activation circuit in MC_2 is deemed to be faulty (*activation check 2*).

5) MC_1 and MC_2 are collectively referred to as *activation-faulty MCs*, and these MCs need fault-recovery.

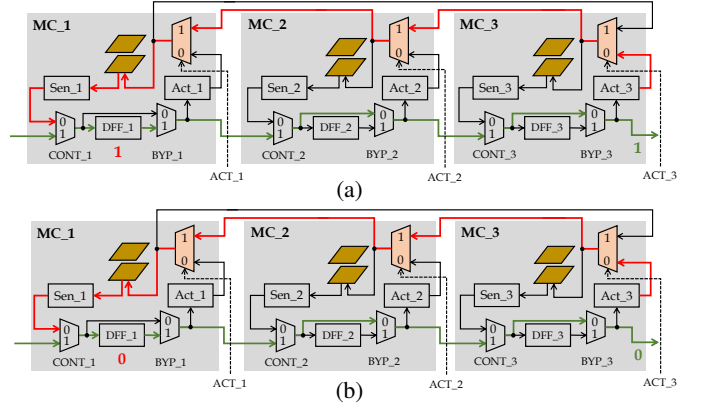


Fig. 4: Illustrations for Stage 4: (a) steps 4) and 5); (b) steps 6) and 7).

Stage 4: MC-activation recoverability test.

1) Select an *activation-faulty MC*, namely MC_1.

2) Configure $ACT_1 = 1$ and $ACT_2 = 0$, such that the bottom plate of MC_1 is connect to Act_2.

3) Because MC_2 is also an *activation-faulty MC*, we need to find another MC as a backup. Configure $ACT_1 = 1$ and $ACT_2 = 1$, and $ACT_3 = 0$, such that the bottom plate of MC_1 is connected to Act_3 (indicated by red arrow in Fig. 4(a)).

4) Shift in pattern “000” and configure $CONT_1 = 0$, such that the bottom plate in MC_1 is driven to 0 V, and the output of Sen_1 is “1”. A rising edge of MC-CLK is applied such that the MC-sensed value of “1” will be stored in DFF_1.

5) Configure $CONT_i = 1$ ($1 \leq i \leq 3$), $BYP_1 = 1$, $BYP_2 = 0$ and $BYP_3 = 0$, such that only DFF_1 is selected. Next, shift out the MC-sensed value and the output bitstream should be “1” (indicated by green arrow in Fig. 4(a)). Suppose the output bitstream is “1”, which passed *activation check 3*.

6) Shift in pattern “111” and configure $CONT_1 = 0$ such that the bottom plate is floating. Then, we apply a 5 V to the top plate. In this case, a 3.3 V is induced in the bottom plate, and the output of the sensing circuit should be 0. A rising edge of MC-CLK is applied such that the MC-sensed value of “0” will be stored in DFF_1.

7) Configure $CONT_i = 1$ ($1 \leq i \leq 3$), $BYP_1 = 1$, $BYP_2 = 0$ and $BYP_3 = 0$, such that only DFF_1 is selected. Next, shift out the MC-sensed value and the output bitstream should be “0” (indicated by green arrow in Fig. 4(b)). Suppose the output bitstream is “0”, which passed *activation check 4*.

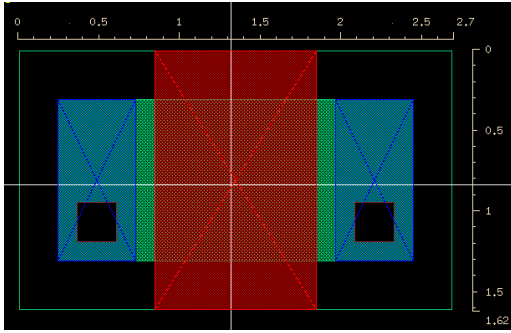
8) Because we pass both *activation check 3* and *activation check 4*, MC_3 can be used to recover the activation circuit of MC_1.

9) A similar process is applied to check if either MC_3 or MC_1 can be used to recover the activation circuit in MC_2.

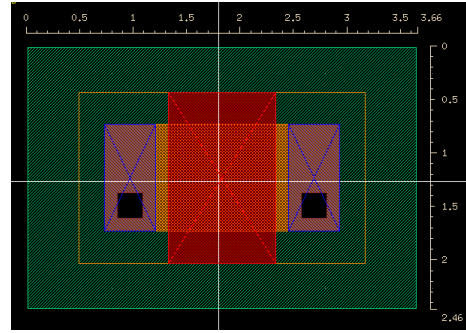
REFERENCES

- [1] G. Wang, *Field-programmable microfluidic test platform for point-of-care diagnostics*. PhD thesis, University of Saskatchewan, 2013.

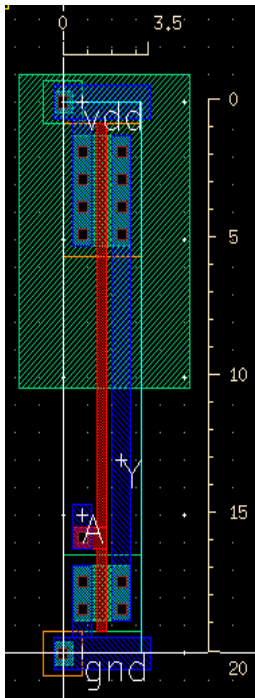
C. Dimensions of Standard Cells



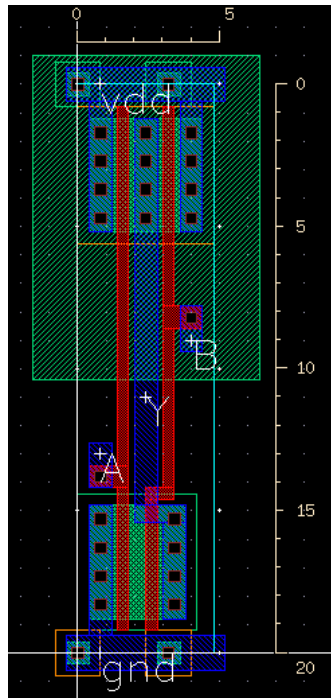
NMOS



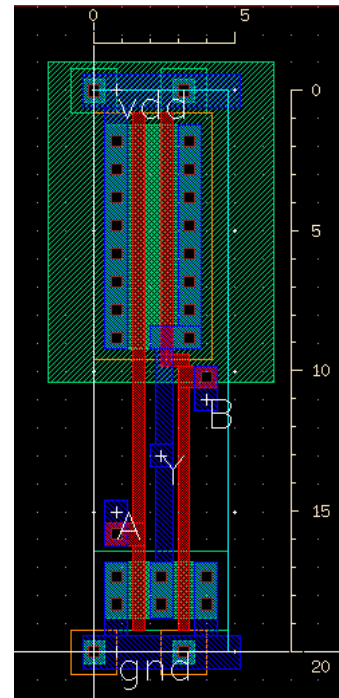
PMOS



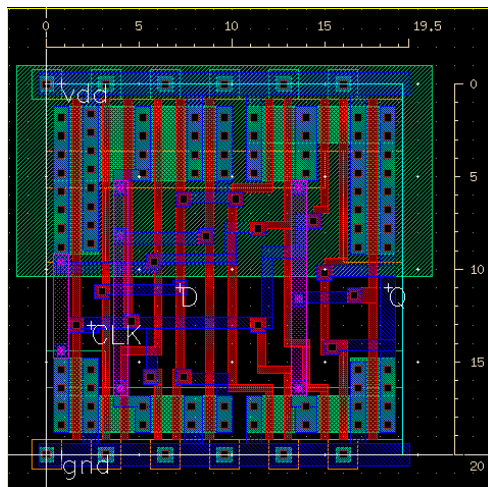
INV_X1



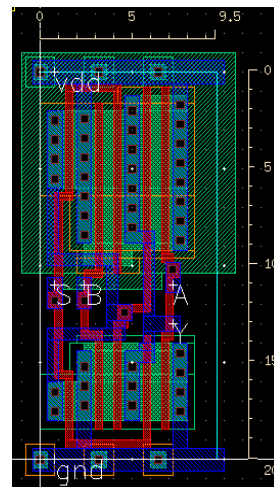
NAND2_X1



NOR2_X1



DFFPOS_X1



MUX2_X1

Unit: μm