

Enhancing the Reliability of MEDA Biochips Using IJTAG and Wear Leveling

Zhanwei Zhong, Tung-Che Liang and Krishnendu Chakrabarty *Fellow, IEEE*

Abstract—**Abstract**—A digital microfluidic biochip (DMFB) enables the miniaturization of 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 control of droplets and seamlessly integrates microelectronics and microfluidics using CMOS technology and a TSMC fabrication process. To ensure that bioassays are carried out on MEDA biochips efficiently, high-level synthesis algorithms have recently been proposed. However, as in the case of conventional DMFBs, microelectrodes are likely to fail when they are heavily utilized, and previous methods fail to consider reliability issues. In this paper, we first present a new microelectrode cell (MC) design such that the droplet-sensing operation can be enabled/disabled for individual MCs. Next, “partial update” and “partial sensing” operations are presented based on an IEEE Std. 1687 IJTAG network design. Finally, wear-leveling synthesis method is proposed to ensure uniform utilization of MCs on MEDA. A comprehensive set of simulation results demonstrate the effectiveness of the proposed hardware design and design automation methods.

I. INTRODUCTION

A digital microfluidic biochip (DMFB) is an example of a lab-on-a-chip that automates the execution of biochemical experiments [1]. Over the past decade, DMFBs have been demonstrated for high-throughput DNA sequencing [2], point-of-care clinical diagnostics [3], and protein crystallization for drug discovery [4]. A DMFB manipulates liquids as discrete droplets of nanoliter and picoliter volumes on a 2D electrode array. This technology has recently been deployed by Genmark for infectious disease testing [5] and by Baebies for disease screening in newborns [6]. These commercialization success stories highlight the emergence of DMFB technology for the marketplace.

However, today’s DMFBs suffer from some key limitations: (1) The droplet size that can be manipulated is fixed; (2) The number of droplet-detection sensors is limited, and real time detection is not available; (3) Fabrication processes are not standardized, and yield and reliability are major concerns. To overcome these limitations, a micro-electrode-dot-array (MEDA) biochip has been proposed [7]–[9]. The MEDA biochip consists of a large number of *microelectrodes* that are arranged in a regular pattern, and these microelectrodes are much smaller than

the electrodes in traditional DMFBs. Multiple microelectrodes are dynamically grouped together to form a fluidic module (i.e., splitter or mixer). MEDA biochips have been fabricated using TSMC 0.35 μm CMOS technology [8]. In MEDA, a 30 V power supply is used to actuate microelectrodes, and 3.3 V is used as the power-supply for the digital circuit that controls the microelectrodes [10].

Under each microelectrode, there is a built-in real-time capacitive-sensing circuit, and this is used to detect the property/location of a droplet. The sensing results from the microelectrodes allow us to develop a high-level synthesis method that can perform real-time error recovery and execute “if-then-else” protocols from biochemistry.

Reliability is important for microfluidic biochips, especially for point-of-care diagnostics, health assessment, and screening for infectious diseases [11]. It has been reported that erroneous bioassay outcomes may result in misleading prescription in point-of-care diagnostics, and such misdiagnosis threaten patients’ lives [12]–[15]. Therefore, it is necessary to take reliability into consideration when we design biochips and generate high-level synthesis results for bioassays [16]–[22].

The first reliability concern is electrode degradation. It is reported in [24] that an electrode is completely degraded after it is charged up to 200 times (due to charge trapping). If an electrode degrades, the force on a droplet decreases, and droplets are likely to get stuck on the charge-trapped electrodes. In some extreme situations, excessive electrode actuation can also lead to dielectric breakdown [25]. In fact, similar biochip degradation concerns are likely to have forced Illumina to halt the sales of its NeoPrep platform in 2017 [26].

Although the above issues have been reported for traditional DMFBs, MEDA biochips also suffer from these problems because a microelectrode in MEDA and an electrode in a traditional DMFB share the same working principle. In a droplet-sensing operation, the bottom plate of the MC is first charged to 3.3 V, and then discharged to 0 V. This process is repeated every

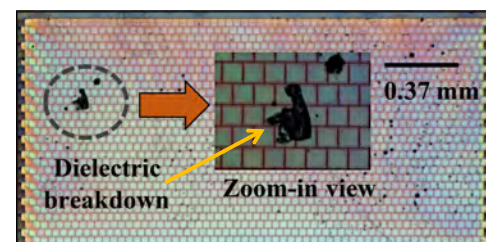


Fig. 1: Illustration of dielectric breakdown in a MEDA biochip [23].

This research was supported in part by the US National Science Foundation under grants CCF-1702596 and ECCS-1914796. A preliminary version of this paper was published in ASP-DAC, 2020.

Z. Zhong is with with Marvell Semiconductor Inc., Santa Clara, CA, USA. T.-C. Liang and K. Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, 27705 USA, e-mail: (zzwzhong698800@gmail.com; tung.che.liang@duke.edu; krish@duke.edu).

Manuscript received May 1, 2020.

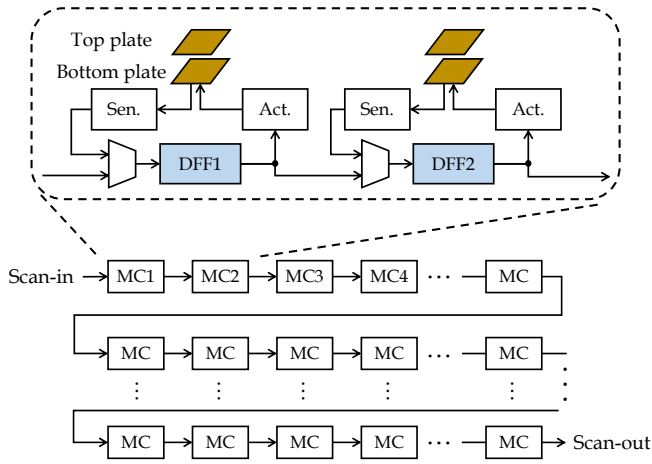


Fig. 3: The logic abstraction of MCs and the scan-chain structure in a fabricated 30×60 MEDA (adapted from [33]).

plate 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 lower, and a value of “1” is stored; otherwise, the DFF stores a value of “0”.

The actuation status of an MC is determined by the value stored in the DFF: a value of “1” indicates MC actuation; otherwise, the MC is not actuated. In the following discussion, we refer to the actuation value for each MC as the *MC-actuation value*. After MC sensing, the result is written to the DFF indicating whether a droplet is present. We refer to the capacitance measured for each MC as the *MC-sensed value*.

B. Scan-Chain Architecture

As shown in Fig. 3, the basic unit of a MEDA biochip is a microelectrode-cell (MC), and it includes five parts: a top/bottom plate, a D Flip-Flop (DFF), an actuation circuit (Act.), and a sensing circuit (Sen.). The MCs are sequentially connected to form a scan chain. In the fabricated chip described in [9], the length of the scan chain is 1800 bits.

In order to perform droplet operations, a group of MCs need to be actuated to form a fluidic module (e.g., a splitter). When an MC is actuated, a high voltage of 30 V is applied to the top plate, and the bottom plate is connected to ground (0 V) to generate a force that can drag a nearby droplet towards the actuated MC [9]. Otherwise, the bottom plate is floating and an induced voltage of 17 V is generated. In this case, no drag force is generated. A value of “1” in the DFF indicates MC actuation; otherwise, the MC is not actuated. A sequence of bits is shifted into the scan chain such that the value in each DFF is updated as intended. We refer to the sequence of shifted bits as an *actuation pattern*.

After MC-actuation, the sensing circuit in each MC measures the capacitance between the top plate and the connected microelectrode. By comparing the sensed capacitance to a preset value, the sensing circuit can determine whether a droplet is present at the microelectrode. The sensing result is then written to the DFF: a value of “1” indicates that a droplet is present;

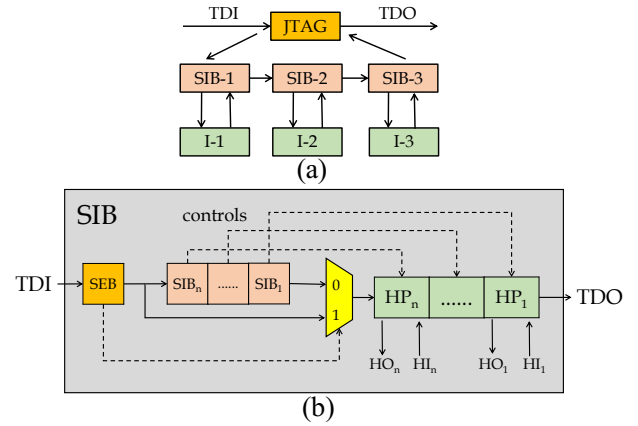


Fig. 4: (a) An example of an IJTAG network and (b) a simplified view of the SIB design.

otherwise, no droplet is present. Next, the 0/1 sensing values of all MCs are shifted out as a sequence of bits, which is referred to as a *sensed pattern*. Note that when an MC-sensed value is latched in a DFF, it overwrites the MC-actuation value. However, the newly-latched MC-sensed value will not change the status of the actuation circuit, because in the MC-sensing process, all actuation circuits are turned off.

C. The IJTAG Network

The IEEE Std. 1687 (IJTAG) [34] provides flexible access to on-chip instruments through the IEEE 1149 JTAG test access port (TAP). It is now being increasingly used for post-silicon validation, production test, fault diagnosis, and fault monitoring [35]–[38]. To provide flexibility of instrument access, a hardware component called the Segment Insertion Bit (SIB) has been introduced [39]. A SIB in the 1687 network is 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 [39].

As shown in Fig. 4(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 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.

A SIB component primarily three parts (Fig. 4):

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

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 [40]. When

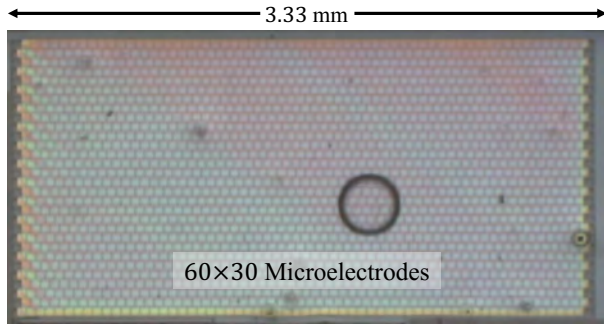


Fig. 5: A droplet on a MEDA biochip.

SEB = 0, the SIB components are included in the primary scan path.

III. MC DESIGN FOR PARTIAL SENSING

In this section, we propose a new MC design such that we can enable/disable the droplet-sensing operation for each MC, i.e., the new MC can support selective sensing for MEDA.

A. Motivation

In today's MEDA biochips, we can manipulate fluids in a fine-grained manner because we can control the actuation status of each MC individually by writing an MC-actuation value to each DFF. However, during a droplet-sensing operation, the bottom plates of all MCs must simultaneously go through an electric discharge-and-charge process to detect which MCs a droplet occupies. In this case, the MCs on MEDA experience rapid degradation because droplet-sensing operation (i.e., discharging and charging process) is carried out every operational cycle.

However, based on a closer inspection of Fig. 5, we can see that a droplet occupies only a small fraction of MCs. Therefore, when we sense the location of droplet, only a small fraction of MCs need to be used. For example, in Fig. 5, all the 1800 MCs are used for droplet-sensing operation per second in the original design. However, the droplet occupies only 40 MCs, and if we perform droplet-sensing operation using only on the occupied MCs, the overall usage of MCs is reduced by 97%, and the lifetime of MCs can be prolonged to a large extent.

The current MC design does not allow us to freely choose which MCs are used for the droplet-sensing operation. Therefore, we present a new MC design that allows us to determine which MCs must be selected and used in a droplet-sensing operation.

B. New MC Design

The schematic of the new MC design is shown in Fig. 6. Compared with the original MC design shown in Fig. 2, we add a two-input OR gate and a DFF to the MC circuit. In general, the value stored in DFF1 is used to determine MC-actuation status when $ACT = 1$ (MC-actuation mode), and is used to enable/disable droplet-sensing functionality when $ACT = 0$ (MC-sensing mode). In addition, the droplet-sensing result is stored in DFF2 when $ACT = 0$ (MC-sensing mode).

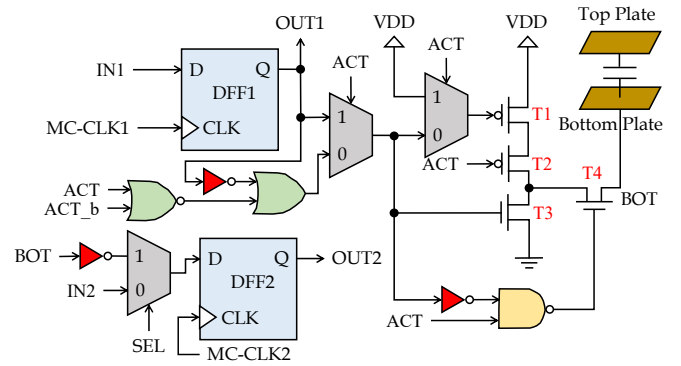


Fig. 6: New microelectrode-cell design for selective sensing.

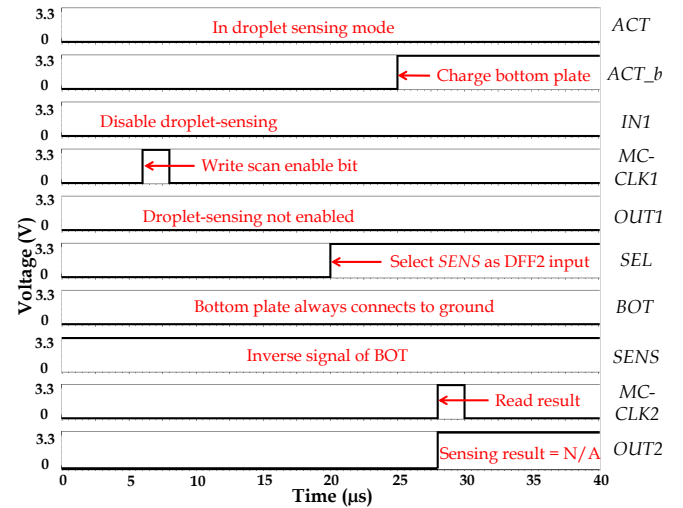


Fig. 7: Simulation results when droplet-sensing operation is disabled.

Suppose an MC is in the sensing mode (i.e., $ACT = 0$), transistors T2 and T4 are always switched on. When a value of "0" is stored in DFF1, the output of the two-input OR gate is equal to "1", which makes T1 switched off and T3 switched on. In this case, the bottom plate is always connected to ground, and discharging and charging do not occur. However, when a value of "1" is stored in DFF1, the status of transistors T1 and T3 are determined by the output of the logic "NOR" of ACT and ACT_B , which is the same as in the conventional MC design. Therefore, the droplet-sensing operation can now be carried out.

After the sensing operation is completed, a rising edge of $MC - CLK2$ signal is applied to DFF2 to store the droplet-sensing result. When a droplet is present, a value of "1" is stored; otherwise, a value of "0" is stored.

C. Design Validation using HSPICE Simulation

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

1) Droplet-sensing disabled. In this operation (illustrated in Fig. 7), a "0" value is first shifted in to the MC using the scan-chain, and the "0" value of $OUT1$ indicates that droplet-sensing

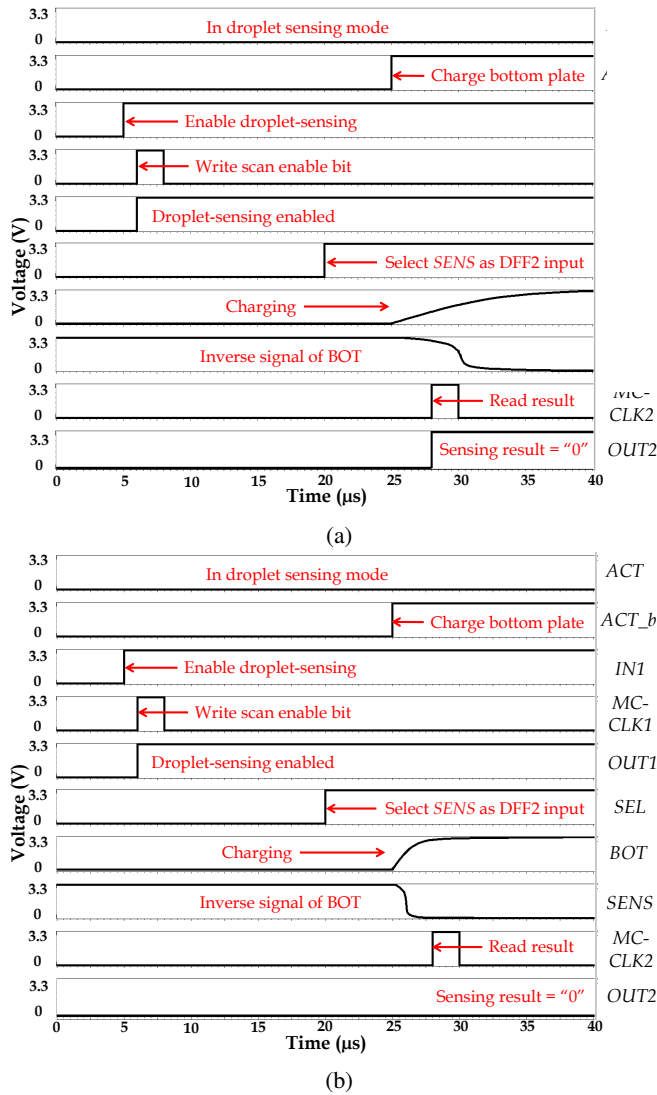


Fig. 8: Simulation results for the droplet-sensing operation: (a) with a droplet, and (b) without a droplet.

operation is *disabled*. Therefore, for the sensing-mode period, the bottom plate is always connected to ground. Although a value of “1” is stored in DFF2 after the rising edge of $MC - CLK2$, this value is marked as “N/A” (not applicable) when it is received by the biochip controller.

2) Droplet-sensing enabled (with a droplet). In this operation (illustrated in Fig. 8(a)), a “1” value is first shifted into DFF1, and the “1” value of $OUT1$ indicates that droplet-sensing operation is *enabled*. As a result, when $ACT_b = 1$, the bottom plate is connected to VDD (3.3 V), and the voltage of net BOT is rising because of capacitor charging. Note that the presence of a droplet increases the dielectric constant, and the capacitor requires longer time to charge. When a rising edge of $MC - CLK2$ is applied, the sensing output $SENS$ is at a high voltage level. Therefore, an MC-sensed value of “1” is store in DFF2, which indicates that a droplet is present.

3) Droplet-sensing enabled (without a droplet). In this operation (illustrated in Fig. 8(b)), a “1” value is first shifted into DFF1, and the “1” value of OUT indicates that droplet-

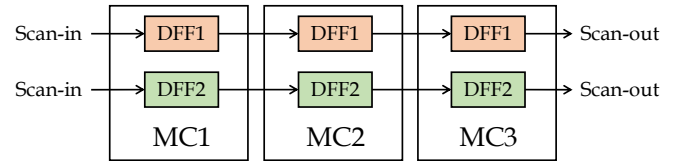


Fig. 9: Scan-chain stitching for the new MC design.

sensing operation is *enabled*. As a result, when $ACT_b = 1$, the bottom plate is connected to VDD (3.3 V), and the voltage of net BOT is rising because of capacitor charging. Without a droplet present, the dielectric constant is smaller and it takes less time to charge the capacitor. When a rising edge of $MC - CLK2$ is applied, the sensing output $SENS$ is at a low voltage level. Therefore, an MC-sensed value of “0” is store in DFF2, which indicates that no droplet is present.

In recent years, we have reported experimental results on fabricated MEDA biochips in collaboration with partners at National Chiao Tung University in Taiwan [23], [29], [41]–[44]. We will continue this collaboration to fabricate the next generation of MEDA biochips based on the new MC design.

D. New Scan-Chain Design

In the original design, each MC has only one DFF. The DFFs from different MCs are serially together to construct a scan-chain architecture; see Fig. 3. However, in the new MC design, there are two DFFs: DFF1 and DFF2. DFF1s from different MCs are serially connected to form a scan-chain. This scan chain is referred to as a *control scan-chain* because it holds the bitstream that controls the MC-actuation status (in the actuation mode), and enable/disable the droplet-sensing operation (in the sensing mode) for each MC. In addition, DFF2s from different MCs are serially connected to form a scan-chain. This scan chain is referred to as a *sensed scan-chain*, because it contains the droplet-sensing results of different MCs.

An example of the new scan-chain design for three MCs is shown in Fig. 9. Three DFF1s are serially connected to form the *control scan-chain* while three DFF2s are serially connected to form the *sensed scan-chain*.

When a bioassay is performed on MEDA with the new MC design and the new scan-chain design, the following steps will be repeated per operational cycle:

Step 1: MC-Actuation Pattern. A bitstream of *MC-actuation values* (i.e., actuation pattern) is shifted into the *control scan-chain* to actuate the corresponding MCs when MEDA is in the actuation mode (i.e., $ACT = 1$).

Step 2: MC-Sensing Enable Pattern. A bitstream of *MC-sensing enable values* is shifted in the *control scan-chain* to enable/disable droplet-sensing operation in the corresponding MCs when MEDA is in the sensing mode (i.e., $ACT = 0$).

Step 3: MC-Sensed Pattern. After rising edge of $MC - CLK2$, a bitstream of *sensed values* (i.e., sensed pattern) is stored in the *sensed scan-chain*. This sensed pattern is then shifted out to obtain the location of the droplet.

Note that the location information of droplets is still available as in the original MEDA architecture; no information is lost and we can still precisely determine the location of a droplet.

The area of each micro-electrode is $50 \mu\text{m} \times 50 \mu\text{m} = 2500 \mu\text{m}^2$ [45], and the area of the new control-circuit design in Fig. 6 is only $2100 \mu\text{m}^2$. In MEDA, each control circuit is placed under the corresponding micro-electrode. Since the area of the new control circuit is less than the area of a micro-electrode, area overhead is not an issue here.

According to the simulation results based on the AMI 0.35 μm process [28], the static and dynamic power consumption values for a 30×60 MEDA biochip using the original design are $0.925 \mu\text{W}$ and 1.14 mW , respectively. When the MC design (i.e., dual scan-chain design) is used, the dynamic power consumption values are $1.416 \mu\text{W}$ and 1.14 mW , respectively, which is still negligible.

IV. PROPOSED IJTAG NETWORK

In this section, we first present the IJTAG network for the MEDA design. Next, partial actuation and partial sensing are introduced based on the new MEDA design.

A. IJTAG Network

In the new design, we first divide the biochip in blocks. For a fabricated 30×60 MEDA biochip, we define each block as a 10×10 rectangle, then the biochip is divided into 18 blocks (from B_1 to B_{18} in Fig. 10(a)). Next, the DFF1s/DFF2s in each block are sequentially connected to form a sub-scan chain (SSC). The SSC1/SSC2 that corresponds to block B_i is referred to as $SSC1_i$ /SSC2 $_i$. Since there are 18 blocks in the biochip, the selected/unselected status of 18 SSC1s/SSC2s are controlled by a 18-bit SIB register; see Fig. 10(b) and Fig. 10(c).

B. Partial Update and Partial Sensing

In today's MEDA design, an actuation pattern with a 1-bit shift is used to update the MC-actuation value for each block. However, according to Section III.D, we find that the MC-actuation pattern and MC-sensing enable pattern are alternately shifted into the scan-chain every cycle. Because the MC-actuation pattern and the MC-sensing enable pattern are highly correlated, we can "write" MC-actuation pattern based on the previous MC-sensing enable pattern, and vice versa.

An example is shown in Fig. 11, in which each green rectangle represents a block. Suppose at operational cycle t , a droplet that resides on Blocks 2, 3, 8, 9 needs to move to the right. To do that, "1"s are written in the DFF1 cells of Blocks 3, 4, 9, 10 in order to actuate the MCs at operational cycle t . After MC-actuation, we need to enable the droplet-sensing operation for Blocks 2, 3, 4, 8, 9, 10. In other words, "1"s should be written in the DFF1 cells of Blocks 2, 3, 4, 8, 9, 10 (i.e., MC-sensing enable pattern). However, because "1"s are already written in the DFF1s of Blocks 3, 4, 9, 10, we only need to write "1"s to the DFF1 cells of Blocks 2 and 8. In operational cycle $t + 1$, the MCs in Blocks 4, 5, 10, 11 need to be actuated. Because the previous MC-sensing enable pattern has written "1"s to the DFF1 cells of Block 2, 3, 4, 8, 9, 10, we only need to write "0"s to the DFF1 cells of Blocks 2, 3, 8, 9, and write "1"s to the DFF1 cells of Blocks 5 and 11.

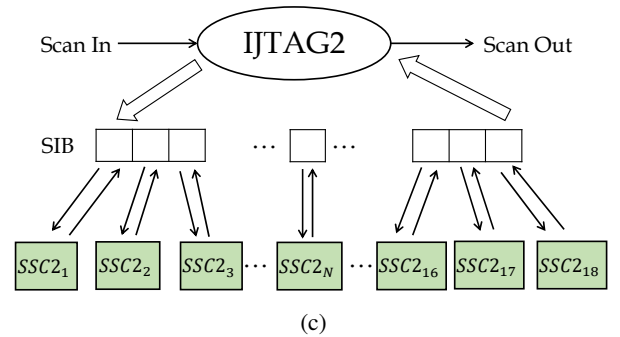
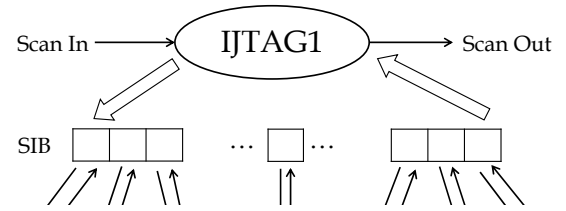
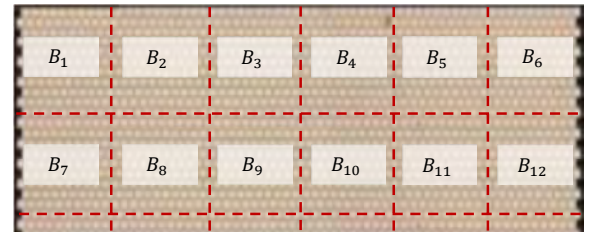


Fig. 10: (a) The floorplan for a 30×60 biochip with 10×10 blocks, (b) the IJTAG network that is used to manage the control scan-chain, (c) the IJTAG network that is used to manage the sensed scan-chain.

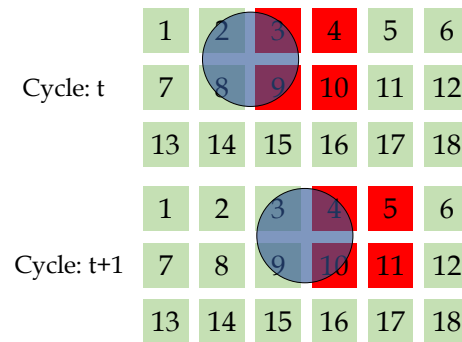


Fig. 11: The MC-actuation status for two consecutive operational cycles.

In summary, when we write in MC-actuation patterns, we first find out the differences between the current MC-actuation pattern and the previous MC-sensing enable pattern. Next, we determine which SSC1s are involved, and use the IJTAG network

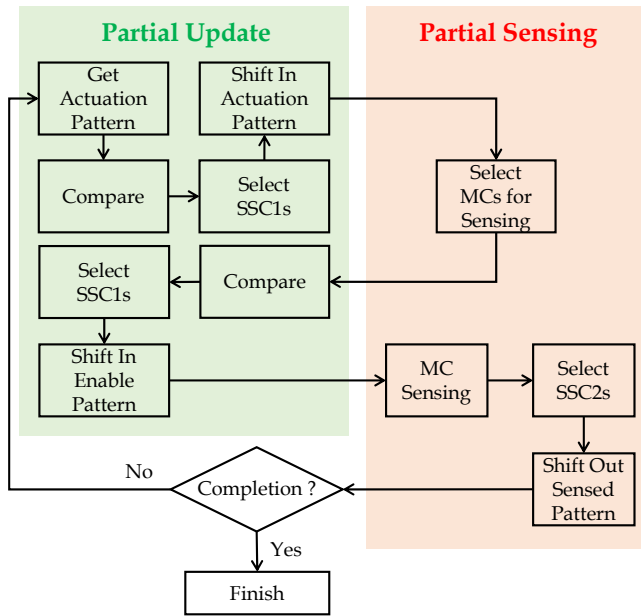


Fig. 12: The control flow for the new MEDA design.

to select the corresponding SSC1s. Finally, the corresponding data-bits are written into the selected SSC1s. Similarly, we can also derive the write-in process for MC-sensing enable pattern based on a previous MC-actuation pattern. Here, the intelligent write-in process of MC-actuation pattern and the MC-sensing enable pattern using the IJTAG network are collectively referred to as *partial update*. Partial update can significantly reduce the number of data-shift operations.

In the original MEDA design, droplet-sensing operations are performed in all MCs to obtain the locations of droplets. However, as discussed in Section III.A, it is unnecessary to perform droplet sensing in all MCs. Instead, we only need to do it in the MCs on which a droplet resides and the MCs that are actuated in the current cycle. After we write-in the MC-sensing enable pattern, droplet-sensing operations are carried out on selected MCs. Following the rising edge of $MC - CLK2$, the sensing results are stored in the selected MCs. In this case, we first determine which SSC2s are involved, and use the IJTAG network to select the corresponding SSC2s. Finally, the MC-sensed patterns from these SSC2s are shifted out. The process of selecting MCs for a droplet-sensing operation and the readout of the MC-sensed pattern are collectively referred to as *partial sensing*. Partial sensing reduces the usage of MCs for droplet-sensing operations.

Based on the above description, the control flow for the MEDA biochip with the new MC design, the new scan-chain design, and the IJTAG network is shown in Fig. 12. It is easy to see that, compared with the control flow for the previous method [46], a “shift enable pattern” procedure has been added to select which MCs are needed to carry out the droplet-sensing operation.

V. HIGH-LEVEL SYNTHESIS BASED ON WEAR-LEVELING

In this section, we present a wear-leveling algorithm to ensure uniform utilization of MCs, with the goal of keeping all the MCs

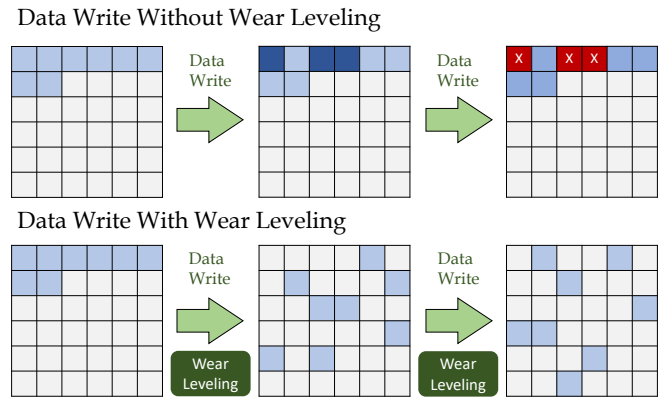


Fig. 13: An illustration of wear leveling in NAND flash memory (adapted from [48]).

“alive” as long as possible.

A. Motivation

The block-aware high-level synthesis method reported in [46] can reduce the overall usage of MCs by up to 80%. However, simulation results presented in Section VI show that the usage distribution for the MCs is not uniform: some of them are rarely used while others are used much more frequently. This usage-distribution problem is also important for data management in NAND flash memory. A NAND flash is an array of equal-sized blocks. Two consecutive writes to the same physical memory location must be interleaved by erasing this location in terms of blocks. However, each memory block can only endure a limit number of data erasures. For example, the MLC-flash block can endure only 5 K cycles of erasure [47]. Therefore, in a workload with both frequently updated data and rarely updated data, some memory blocks wear out while other blocks remain fresh.

To address this problem, a technique called *wear leveling* has been utilized [47], [49], [50]. Wear leveling is a policy that can evenly erase all flash memory blocks, with the goal of keeping all the blocks alive as long as possible. An illustrative example is shown in Fig. 13. The first row shows the situation in which some of the memory blocks are excessively used so that these blocks are “dead” at an early stage of the product life cycle. However, in the second row, wear leveling is used to average the usage of memory blocks so that the NAND flash memory device can last for a longer period of time.

Another motivation for the proposed approach comes from the fact that the assay completion times are not completely predictable and multiple runs of some bioassay steps are often needed. According to [1], a control-flow graph (CFG) is used to carry out gene-expression analysis protocol on a digital microfluidic biochip. Some bioassay steps such as purification have to be repeated until the results fall in an acceptable range. In this case: (1) if wear leveling is not used, some electrodes will be completely degraded before we get acceptable result; (2) if wear leveling is used, the desired results can be obtained with higher likelihood because the lifetime of the biochip is prolonged significantly. This is of considerable value, even for disposable microfluidic biochips.

B. Problem Formulation

The problem statement for high-level synthesis based leveling is as follows:

Input: (1) The sequencing graph $G = (V, E)$ for the where V represent the set of fluidic modules and E the dependencies between fluidic modules [29]; (2) The biochip library, which includes the type, size and correct execution time of on-chip fluidic functional modules; size of the MEDA biochip; (4) The IJTAG network of MEDA biochips.

Output: Schedule of operations, module placement, droplet routing pathways.

Objective: Minimize the cost function C given by:

$$C = \sqrt{\frac{\sum_{i=1}^N U(MC_i) - \bar{U}}{N}}$$

where $U(MC_i)$ is the number of times MC_i is utilized, \bar{U} is the average usage for all MCs, and N is the total number of MCs on a MEDA biochip. This cost function is essentially the standard deviation of the usage of different MCs, and it describes how evenly the usage is distributed among MCs. A smaller value of the cost function indicates a more desirable outcome for the wear-leveling technique.

The definition of $U(MC_i)$ and \bar{U} are as follows:

$$\begin{aligned} U(MC_i) &= N_a(MC_i) + N_s(MC_i) \\ \bar{U} &= \frac{\sum_{i=1}^N U(MC_i)}{N} \end{aligned} \quad (3)$$

where $N_a(MC_i)$ is the total number of times MC_i is actuated, $N_s(MC_i)$ is the total number of times MC_i is used for droplet-sensing operations, and N is the total number of MCs in a MEDA biochip.

C. Young Pool and Old Pool

When a MEDA biochip is being used, the control software keeps track of the utilization of all the MCs. We define the usage metric for B_i as the summation of the usage of MCs in B_i , which can be expressed as:

$$U(B_i) = \sum_{MC_j \in B_i} U(MC_j) \quad (4)$$

We refer to a block with a small usage metric as a “young” block. On the other hand, we refer to a block with a larger usage metric as an old block. The key idea underlying the high-level synthesis based on wear leveling is that we attempt to utilize “young” blocks for module placement and droplet routing, and avoid using “old” blocks.

In order to categorize a block as either “young” block or “old” block, we utilize two data structures, referred to as the *young pool* and the *old pool*. As shown in Fig. 14, the young pool and the old pool are two priority queues. Each element in a priority queue is a block, and these blocks are sorted according to the usage metric value.

When a MEDA biochip is first powered on, the usage metric for each block is 0. Half of the blocks are randomly assigned to the young pool and the other half are assigned to the old

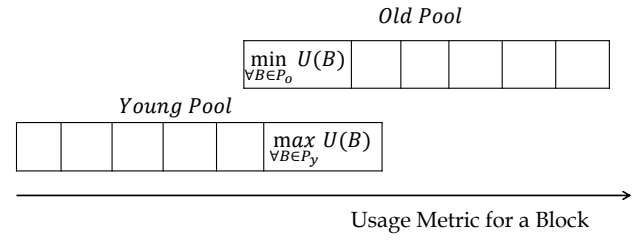


Fig. 14: An illustration for the young pool and the old pool.

pool. When bio-protocols are executed on a MEDA biochip, the usage metric for blocks will change in different ways. We consider two decisions as part of the synthesis step: (1) some of the blocks in the young pool are heavily used and should be moved to the old pool, and (2) some of the blocks in the old pool are rarely used and should be moved to the young pool.

According to the above discussion, we have the following pool-update policy:

Young-Old Swapping Condition. Suppose that P_y is the set of blocks in the young pool, and P_o is the set of blocks in the old pool. Upon the completion of a cycle of MC-actuation and MC-sensing operation, check the following condition:

$$\max_{B \in P_y} \{U(B)\} - \min_{B \in P_o} \{U(B)\} > TH \quad (5)$$

where $\max_{B \in P_y} \{U(B)\}$ is the block with the maximum usage metric in the young pool, $\min_{B \in P_o} \{U(B)\}$ is the block with the minimum usage metric in the old pool, and TH is a user-defined threshold.

If the condition given by (5) holds, block $\max_{B \in P_y} \{U(B)\}$ is no longer considered to be a “young” block and $\min_{B \in P_o} \{U(B)\}$ is no longer considered to be an “old” block. Therefore, we perform the following procedure:

Step 1. Move block $\max_{B \in P_y} \{U(B)\}$ from the young pool to the old pool.

Step 2. Move block $\min_{B \in P_o} \{U(B)\}$ from the old pool to the young pool.

Step 3. Check condition given by (5). If this condition is true, repeat Step 1 to Step 3. Otherwise, stop.

Note that the above procedure is guaranteed to terminate in $\lfloor N/2 \rfloor$ iterations, where N is the total number of blocks in a MEDA biochip. The block with the minimum usage metric in the young pool is $\min_{B \in P_y} \{U(B)\}$ while the block with the maximum usage metric in the old pool is $\max_{B \in P_o} \{U(B)\}$. The worst-case situation occurs when the following condition is satisfied:

$$\min_{B \in P_y} \{U(B)\} - \max_{B \in P_o} \{U(B)\} > TH \quad (6)$$

In this situation, a total of $\lfloor N/2 \rfloor$ swapping steps are needed.

D. Wear-Leveling High-Level Synthesis Algorithm

The pseudocode for the wear-leveling high-level synthesis algorithm is shown in Fig. 15. The input to this algorithm is the sequencing graph, and a trade-off parameter α . In a sequencing graph, the vertex represents a droplet operation, and the edge between two vertices represents the droplet-operation dependency. Droplet operations in the sequencing

Algorithm 1 high_level_synthesis(graph, α)**Input:** The DAG graph, and the trade-off parameter α ;**Output:** The placement and routing for each droplet operation.

```

1: dict_loc := {};
2: dict_route := {};
3: time := 0;
4: lst_wait := [];
5: lst_ready := lst_op_dispende;
6: lst_process := [];
7:
8: while len(lst_ready) + len(lst_process) > 0 do
9:   for op in sort_list(lst_ready,  $\alpha$ ) do
10:    min_cost, lst_loc := get_cost_wl(op);
11:    for loc in lst_loc do
12:      while  $\beta \leq 1$  do
13:        best_loc := find_best_loc(lst_loc,  $\beta$ );
14:        route := router(best_loc);
15:        if route != None then break;
16:        else  $\beta := \beta + 0.1$ ;
17:      if route != None then
18:        dict_loc[node] := best_loc;
19:        dict_route[node] := route;
20:        break;
21:      if route = None then
22:        exit; // synthesis fails
23:
24:    time := time + 1;
25:    update_pool(young_pool, old_pool);
26:
27:    for op in lst_process do
28:      if op.finish_time <= time then
29:        lst_process.remove(op);
30:        update_list(lst_wait, lst_ready);
31:
32: return dict_loc, dict_route;

```

Fig. 15: The pseudocode for high-level synthesis.

graph are scheduled using the list-scheduling algorithm [51]–[53]. Compare with the traditional list-scheduling method, we have the following improvements:

- We have added a “young pool” and a “old pool” data structures to keep track of the MC-usage information for each block.
- In the module-placement step, we have develop a wear-leveling cost function $C_{wl}(loc)$ so as to place a fluidic module in the block that is still “young”.
- We also developed a module-placement cost function $C_{mp}(op)$ such that the droplet routes can be reduced.
- In every clock cycle, the “young pool” and “old pool” data structures will be updated according to the *Young-Old Swapping Condition*.

In order to implement the list-scheduling algorithm, three list are first created: waiting list, ready list and processing list (Line 4-5). At the beginning of the algorithm, all the droplet-dispensing operations are in the ready list while the remaining operations in the sequencing graph are placed in the waiting list.

We first check if the ready list and the processing list are empty (Line 8). If both of them are empty, it indicates that the

scheduling has been completed. Otherwise, we use the function $sort_list(lst_ready, \alpha)$ to sort the droplet operations in the ready list in a descending order according to the priority for each operation (Line 9), which is computed using the following equation:

$$P(op) = \alpha \times D(op) + (1 - \alpha) \times (L_{max} - L(op)) \quad (7)$$

where $D(op)$ is the out-degree of op in the sequencing graph, $L(op)$ is the level of op in the sequencing graph using topological sort, L_{max} is the height of the sequencing graph, and α is a trade-off parameter ($0 \leq \alpha \leq 1$).

For each operation op , we use the function $get_cost_wl(op)$ to obtain the minimum wear-leveling cost and store the corresponding locations in lst_loc (Line 10). The computation of the wear-leveling cost for loc can be expressed as follows:

$$C_{wl}(loc) = \frac{\sum_{B_i \in P_y} U(B_i) + \sum_{B_i \in P_o} C}{N} \quad (8)$$

where N is the total number of blocks occupied by the module placed at loc , $U(B_i)$ is the sum of MC usage in block B_i , P_y is the set of blocks in the young pool, P_o is the set of blocks in the old pool, and C is a large constant number that is set to 999,999 in our code implementation. The location with a smaller value of $C_{wl}(loc)$ is more favorable because it indicates that the module at location loc occupies more young blocks in the young pool.

After we obtain lst_loc , the function $find_best_loc(lst_loc, p)$ is used to carry out a linear search for the locations in lst_loc , and obtain the location (i.e., $best_loc$) that has the minimum module-placement cost (Line 13), which can be expressed as follows:

$$C_{mp}(op) = (1 - \beta) \times \sum_{p \in \mathbb{P}} Man(loc(p), loc(op)) + \beta \times \sum_{c \in \mathbb{C}} overlap(Box(c), Box(op)) \quad (9)$$

where op is the droplet operation being evaluated, \mathbb{P} is the set of predecessor of op , $Man(A, B)$ is the Manhattan distance from location A to location B, $loc(p)$ and $loc(op)$ are the location of predecessor p and the location of operation op , respectively. A lower value of the first term indicates a shorter length for droplet routes and thus implies that fewer MCs are used.

In the second term, \mathbb{C} is the set of operations in the ready list whose module placements and the routes from their predecessors are determined. For an operation c , the source of the route is the location of its predecessor and the sink of the route is the target module placement. The bounding box for operation c is defined as the rectangle area whose diagonal corners are the source and sink of the route for operation c . Here, we use $BB(c)$ and $BB(op)$ to represent the bounding boxes for operation c and operation op , respectively. $overlap(BB(c), BB(op))$ is the overlapping area between $BB(c)$ and $BB(op)$.

Finally, β is a trade-off parameter. In the scheduling algorithm, β is initially 0, which implies that reducing the length of the route is the first priority. However, prioritization along can lead to route congestion for a later route. Therefore, an increase in β can alleviate the problem of routing congestion and make droplet routing easier.

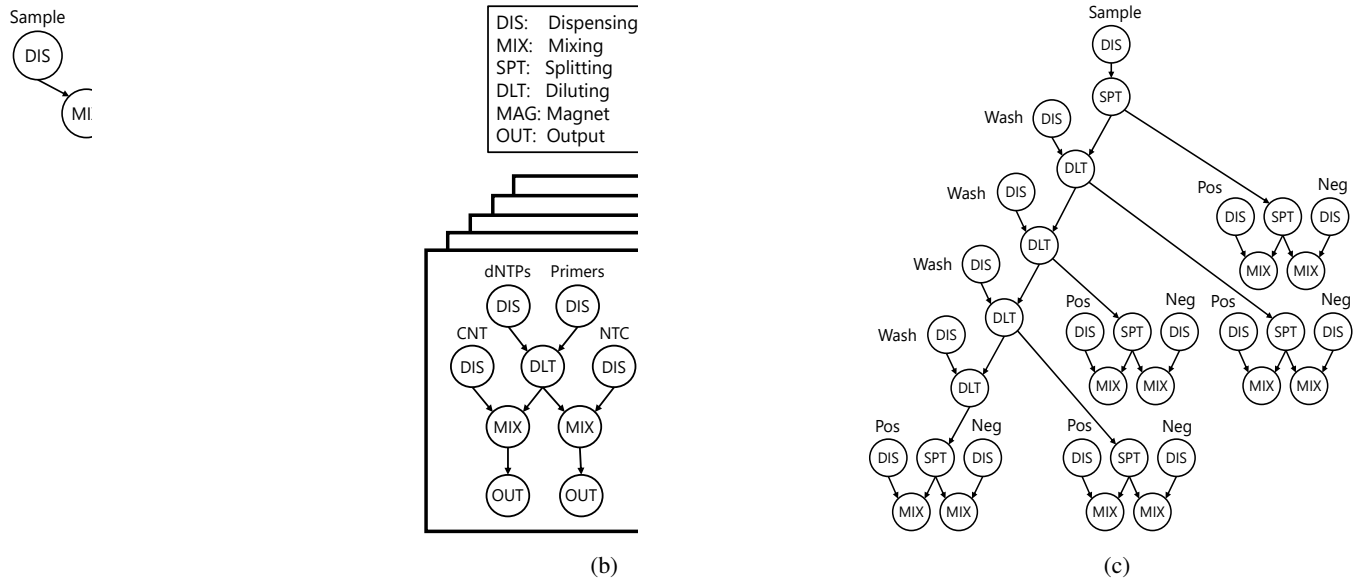


Fig. 16: (a) The CEP benchmark, (b) the master-mix benchmark, and (c) the serial-dilution benchmark.

After we select the target location for an operation, we use the function *router* to determine the routes that come from its predecessors (Line 14). Droplet routes in a MEDA biochip can be extended to a 3D space. The x axis and the y axis correspond to the 2D plane of MEDA, and the z axis corresponds to the operational cycles. We use the 3D *A**-algorithm [54] to obtain the shortest route from a parent operation to a child operation. In this way, the number of blocks used for partial update can be kept small. This process is repeated until we determine the locations and routes for all the operations in the ready list.

If a route cannot be found from a parent operation to the target location corresponding to a child operation, we increase the value of p (Line 16), find a new target location for the operation op , and carry out re-routing. This process is repeated until we successfully find a route from a parent operation to the target location (Lines 17-20).

The time (i.e., operation cycle) is then increased by one unit, and the young/old pool is updated accordingly (Lines 24-25). After that, we check if an operation in the processing list has finished. If this is the case, the operation is removed from the processing list (Lines 28-29). Finally, we use the function *update_list(lst_wait, lst_ready)* to move the operations in the waiting list, whose parent operations have been finished, to the ready list (Line 30).

The high-level synthesis algorithm can be evaluated using the cost function in Equation (2). With different values of the trade-off parameter α , we will obtain different synthesis solutions and different cost. We carry out a linear search on α ($0 \leq \alpha \leq 1$) and select the synthesis solution with the minimum cost.

VI. SIMULATION RESULTS

In this section, we use three real-life benchmarks, namely CEP, serial dilution, and master mix [55] for evaluation; see Fig. 16. CEP is a combination of three small bioassays: cell lysis, mRNA extraction, and mRNA purification. In the simulation setup, the size of the biochip is 30×60 , and the block size for the IJTAG network is set to 10×10 . The dispensing

TABLE I: The number of times benchmark bioassays can run on a MEDA biochip for the different methods.

Methods	CEP	Master-Mix	Serial-Dilution
M-1	2	4	1
M-2	3	4	1
M-3	12	7	5
M-4	8	6	3

time for the reservoir is 2 s. The mixing time is 3 s for an 8×8 mixer, the splitting time is 2 s for a 4×4 splitter, and the dilution time is 5 s for an 8×8 diluter [29]. The threshold TH in the wear-leveling algorithm is set to 50.

Note that this paper does not consider wash operations during bioassay execution, because wash droplets are considerably larger than sample or reagent droplets and their movement is considerably slower. Moreover, it is a major challenge to ascertain whether a wash droplet was able to successfully clean the suspected contaminated site. The preferred solution today is to carry out electro-wetting and the associated droplet manipulations in a film of inert oil (e.g., silicone oil) [56]–[58]. This silicon-oil based solution has been adopted in some commercial products (e.g., ePlex from Genmark and SEEKER from Babies) [57], [59].

A. Simulation Results

In order to quantify the benefits introduced by the IJTAG network and wear-leveling high-level synthesis, we compare the following four methods:

M-1: The high-level synthesis method in [29] for the baseline MEDA design.

M-2: The block-aware high-level synthesis method in [46] applied to a MEDA biochip with an IJTAG network (partial actuation/sensing operations are utilized).

M-3: The proposed wear-leveling high-level synthesis method based on a MEDA biochip with two IJTAG network for two scan-chains (partial update/sensing operations are utilized).

TABLE II: Simulation results for CEP bioassay (run for 2 times), master-mix bioassay (run for 4 times) and serial-dilution bioassay (run for 1 time).

Methods	Completion Time (s)	Data-Shift Bits	$U(MC)$ min	$U(MC)$ max	$U(MC)$ mean	$U(MC)$ σ
CEP Benchmark						
M-1	51	183.6 K	51	198	58	18
M-2	54	17.3 K	0	136	7	17
M-3	58/60/64	30.5 K	0	32	8	6
Master-Mix Benchmark						
M-1	38	136.8 K	38	192	83	45
M-2	40	42.4 K	0	200	32	41
M-3	44/45/49	74.3 K	0	152	37	25
Serial-Dilution Benchmark						
M-1	81	291.6 K	81	195	73	16
M-2	90	44.9 K	0	142	7	20
M-3	98/98/98	80.7 K	8	40	10	8

M-4: The proposed wear-leveling high-level synthesis method based on a MEDA biochip with one IJTAG network (only partial update is utilized). In every clock cycle, all MCs need to carry out droplet-sensing operation because selective sensing is not available.

According to [24], an electrode is completely degraded when it is actuated 200 times. Here, we also assume that the MEDA biochip is deemed to be not usable if any MC is used for over 200 times. Table I shows the number of times we can run CEP, Master-Mix and Serial-Dilution bioassays using M-1, M-2, M-3 and M-4 with a limit of 200 actuations of an MC. The results show that M-3 provides the best results; M-4 also provides significant improvements over M-1 and M-2, but not as much as M-3 because partial sensing is not utilized in this method.

In the next part of simulation setup, each benchmark bioassay is performed on the MEDA biochip for a total of N times. Note that the value of N is bioassay- and method-specific; see Table I. This is because the bioassays have different schedules and completion times, and they require different actuation patterns as well as various numbers of actuation for the MCs. We use the following metrics to evaluate M-1, M-2 and M-3:

(1) **Bioassay Completion Time.** For M-1 and M-2, the completion time is the same for each run. However, for M-3, the bioassay completion time varies for each run because wear-leveling high-level synthesis will generate different module placements and droplet routes to average the usage of MCs. Therefore, in Table II, we present the minimum/average/maximum completion time for M-3.

(2) **Amount of Data Shift.** This parameter is the average number of bits that are needed to shift in the MEDA scan chain. In M-1, 3600 data-bits have to be shifted into the scan chain per operational cycle. Therefore, the total number of shifted bits is $3600 \times CT$, where CT is the completion time. However, with the help of the IJTAG network design, a much smaller amount of data shift is needed.

(3-5) **Minimum/Maximum/Mean MC Usage.** The mini-

mum/maximum/average number of times an MC is used, among all MCs after N runs of a benchmark bioassay.

(6) **Standard Deviation (σ) of MC Usage.** This metric describes how evenly the usage is distributed among MCs. A smaller value of σ indicates a more uniform usage distribution for all the MCs.

As shown in Table II, in the baseline method M-1, because MC-actuation patterns/MC-sensed patterns are shifted in/out using the original scan-chain architecture, the number of data-shift operations is very high. In addition, because all MCs on MEDA are utilized for the droplet-sensing operation, the overall usage of all the MCs is also very high.

Compared with M-1, due to the IJTAG network design, the number of data-shift operations is significantly reduced, by up to 70%. In addition, because M-2 utilizes a block-level partial sensing technique, the overall usage of all the MCs is also reduced by up to 90% (see the column of average MC usage). However, because both M-1 and M-2 do not attempt to ensure the uniform usage of MCs, we can see that the standard deviation value is large for both M-1 and M-2.

According to Table II, a major benefit introduced by M-3 is that all the MCs are more evenly utilized, which can be inferred from a much lower value of the standard deviation. However, to average out the MC-usage value, the completion time and the average MC usage is slightly increased compared to M-2.

B. Histogram and Distribution of MC Usage

Fig. 17 shows the histograms of the MC usage for the three benchmark bioassays when methods M-2 and M-3 are used. If wear leveling is not used (M-2), according to Fig. 17(a), we can see that the usage value of most MCs are very small (less than 20), but some MCs are used for many times (e.g., 200 times). This indicates that, if wear-leveling technique is not used, some MCs will be excessively actuated, which will lead to an early failure of the MEDA biochip.

In contrast to M-2, if wear-leveling technique is used (M-3), there is a considerable reduction in the number of MCs that are frequently actuated. For example, after running the CEP bioassay for three time, the maximum MC-usage value for M-2 is nearly 200, while the maximum MC-usage value for M-3 is only 47. The reduction of the maximum MC-usage value can significantly prolong the lifetime of a MEDA biochip.

Fig. 18 shows the MC-usage value across the 2D plane. Each dot on the 2D plane is geometrically one-to-one mapped to each MC on a fabricated 30×60 MEDA biochip. According to Fig. 18(b), with wear leveling, all the MCs are evenly used. However, without wear leveling, we can see from Fig. 18(a) that the MCs in some location are frequently used while the MCs in other locations are rarely used.

C. Discussion of Parameters

In the previous sections, we have presented the simulation results using a threshold TH of 50 and a block size BL of 10×10 . In this subsection, we will explore how these two parameters affect the performance of M-3.

As shown in Table III, the bioassay completion time varies for each run because method M-3 will generate different module

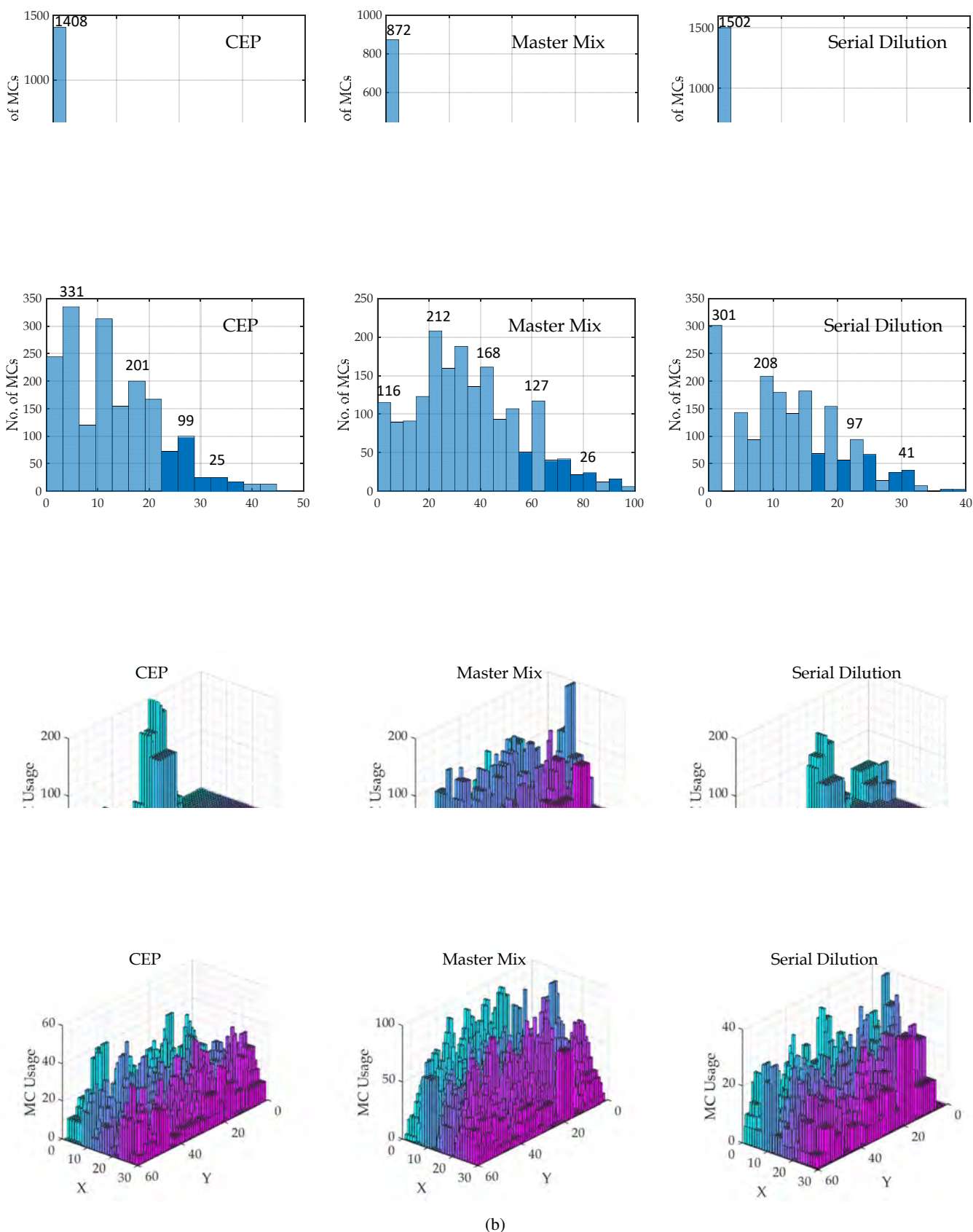


Fig. 18: Distribution of the MC-usage metrics after running CEP bioassay for 3 times ($N = 3$), Master-Mix bioassay for 4 times ($N = 4$) and Serial-Dilution bioassay for once ($N = 1$) when methods (a) M-2 and (b) M-3 are used. The x-y plane represents the MEDA biochip, and the z-axis represents the MC-usage metric.

3

4

1

TABLE III: Simulation results for CEP bioassay (run for 12 times), master-mix bioassay (run for 7 times) and serial-dilution bioassay (run for 5 time) using M-3.

Threshold	Block Size	Completion	Data-Shift	$U(MC)$	$U(MC)$
TH	BL	Time (s)	Bits	mean	σ
CEP Benchmark					
50	10×10	58/60/64	30.5 K	49	29
100	10×10	57/60/64	30.4 K	47	32
200	10×10	55/59/65	30.1 K	46	33
50	5×5	57/60/64	36.5 K	53	27
Master-Mix Benchmark					
50	10×10	44/45/49	74.3 K	65	35
100	10×10	44/44/49	74.5 K	64	36
200	10×10	42/44/51	73.9 K	62	38
50	5×5	44/46/50	78.3 K	68	33
Serial-Dilution Benchmark					
50	10×10	96/98/103	291.6 K	53	27
100	10×10	95/98/104	291.4 K	52	28
200	10×10	93/96/105	290.9 K	51	29
50	5×5	97/98/103	300.6 K	55	25

placements and droplet routes to average the usage of MCs. Therefore, in the third column of Table III, we present the minimum/average/maximum completion time for M-3. We can see that under different combinations of parameters, the bioassay completion times have very small variance. When BL is set to 10×10 , the number of data-shift bits is nearly the same for different values of TH . However, when BL is changed from 10×10 to 5×5 , the number of data-shift bits has a noticeable increment because more configuration bits have to be shifted into the SIB registers.

When BL is 10×10 , a 30×60 MEDA biochip has a total of 18 blocks, and each IJTAG network (see Fig. 10) has a SIB register of 18 bits. In each operational cycle, a total of $18 \times 2 = 36$ configuration bits have to be shifted into the SIB registers. However, when BL is set to 5×5 , a 30×60 MEDA biochip has a total of 72 blocks. This means that each IJTAG has a SIB register of 72 bits, and a total of $72 \times 2 = 144$ configuration bits have to be shifted into the SIB registers in each operational cycle.

From Table III, we also see that when TH is increased from 50 to 200, the overall usage of MCs decreases slightly while the standard deviation (σ) of MC usage increases. In addition, when BL is change to 5×5 , the standard deviation of MC usage is reduced noticeably while the overall MC usage increases. Based on these two inspections, we conclude that there is a trade-off between the overall wear-leveling performance and the overall MC usage. A better wear-leveling outcome can be achieved at a cost of higher overall MC usage overhead.

In practice, for each value of TH , the user can run simulation and determine the number of bioassays that can be run on the MEDA biochip (before any one MC is used for more than 200 times). In order to set an appropriate value of TH , the user can carry out a linear search on TH , and find out the TH value that

yields the maximum number of runs for the target bioassay.

VII. CONCLUSION

We have addressed some key reliability challenges associated with today's MEDA biochip designs. We have presented a new MC design, a new scan-chain design, and the corresponding IJTAG network design for MEDA. Based on the new hardware solution, we have presented a partial update operation, a partial sensing operation, and the overall control flow. Following this, we have introduced a high-level synthesis method based on wear leveling to ensure uniform utilization of MCs. Simulation results for three representative biochips have demonstrated the effectiveness of the proposed method.

REFERENCES

- [1] M. Ibrahim *et al.*, "Cyber-physical digital-microfluidic biochips: Bridging the gap between microfluidics and microbiology," *Proc. of the IEEE*, vol. 106, no. 9, pp. 1717–1743, 2017.
- [2] H. Kim *et al.*, "Automated digital microfluidic sample preparation for next-generation DNA sequencing," *Journal of the Association for Laboratory Automation*, vol. 16, no. 6, pp. 405–414, 2011.
- [3] C. D. Chin *et al.*, "Commercialization of microfluidic point-of-care diagnostic devices," *Lab on a Chip*, vol. 12, no. 12, pp. 2118–2134, 2012.
- [4] A. P. Aijian *et al.*, "Fluorinated liquid-enabled protein handling and surfactant-aided crystallization for fully in situ digital microfluidic MALDI-MS analysis," *Lab on a Chip*, vol. 12, no. 14, pp. 2552–2559, 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'15*, 2015.
- [10] Y. Ho *et al.*, "Design of a micro-electrode cell for programmable lab-on-CMOS platform," in *ISCAS'16*, pp. 2871–2874, 2016.
- [11] K. Chakrabarty, "Towards fault-tolerant digital microfluidic lab-on-chip: defects, fault modeling, testing, and reconfiguration," in *BioCAS'08*, pp. 329–332, 2008.
- [12] S. S. Ali *et al.*, "Security assessment of cyberphysical digital microfluidic biochips," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 3, pp. 445–458, 2015.
- [13] T.-C. Liang *et al.*, "Execution of provably secure assays on meda biochips to thwart attacks," in *ASPDAC'19*, pp. 51–57, 2019.
- [14] R. Kurita *et al.*, "Microfluidic device integrated with pre-reactor and dual enzyme-modified microelectrodes for monitoring in vivo glucose and lactate," *Sensors and Actuators B: Chemical*, vol. 87, no. 2, pp. 296–303, 2002.
- [15] V. Srinivasan *et al.*, "Droplet-based microfluidic lab-on-a-chip for glucose detection," *Analytica Chimica Acta*, vol. 507, no. 1, pp. 145–150, 2004.
- [16] T.-W. Huang *et al.*, "Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips," in *ICCAD*, pp. 448–455, 2011.
- [17] Y.-H. Chen *et al.*, "A reliability-oriented placement algorithm for reconfigurable digital microfluidic biochips using 3-D deferred decision making technique," *TCAD*, vol. 32, no. 8, pp. 1151–1162, 2013.
- [18] S.-T. Yu *et al.*, "Reliability-driven chip-level design for high-frequency digital microfluidic biochips," *TCAD*, vol. 34, no. 4, pp. 529–539, 2015.
- [19] L. Shao *et al.*, "Lookup table based fast reliability-aware sample preparation using digital microfluidic biochips," *TCAD*, 2019.

- [20] G.-R. Lu *et al.*, "Reliability hardening mechanisms in cyber-physical digital-microfluidic biochips," *JETC*, vol. 14, no. 3, pp. 1–22, 2018.
- [21] Z. Li *et al.*, "Reliability-driven pipelined scan-like testing of digital microfluidic biochips," in *ATS*, pp. 57–62, 2014.
- [22] A. Singla *et al.*, "Reliability analysis of mixture preparation using digital microfluidic biochips," *TCAD*, vol. 38, no. 4, pp. 654–667, 2018.
- [23] Z. Li *et al.*, "Structural and functional test methods for micro-electrode-dot-array digital microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 37, no. 5, pp. 968–981, 2017.
- [24] C. Dong *et al.*, "On the droplet velocity and electrode lifetime of digital microfluidics: voltage actuation techniques and comparison," *Microfluidics and Nanofluidics*, vol. 18, no. 4, pp. 673–683, 2015.
- [25] A. I. Drygiannakis *et al.*, "On the connection between dielectric breakdown strength, trapping of charge, and contact angle saturation in electrowetting," *Langmuir*, vol. 25, no. 1, pp. 147–152, 2009.
- [26] "Illumina Drops NeoPrep." <http://omicsomics.blogspot.com/2017/02/illumina-drops-neoprep.html>, 2020 [online].
- [27] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.
- [28] "AMI 0.35 μm PDK." <https://vlsiarch.ecen.okstate.edu/flow>, 2019.
- [29] Z. Li *et al.*, "High-level synthesis for micro-electrode-dot-array digital microfluidic biochips," in *Annual Design Automation Conference (DAC)*, 2016.
- [30] C. F. Hawkins *et al.*, "Test and reliability: Partners in IC manufacturing, part 1," *IEEE Design & Test of Computers*, no. 3, pp. 64–71.
- [31] G. Wang, *Field-programmable microfluidic test platform for point-of-care diagnostics*. PhD thesis, University of Saskatchewan, 2013.
- [32] M. G. Pollack *et al.*, "Electrowetting-based actuation of liquid droplets for microfluidic applications," *Applied Physics Letters*, vol. 77, no. 11, pp. 1725–1726, 2000.
- [33] L. Zhang *et al.*, "Built-in self-diagnosis and fault-tolerant daisy-chain design in MEDA biochips," in *ITC'18*, 2018.
- [34] IEEE Standard Committee *et al.*, "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Standard*, 2014.
- [35] R. Cantoro *et al.*, "Automatic generation of stimuli for fault diagnosis in IEEE 1687 networks," in *IOLTS'16*, pp. 167–172, 2016.
- [36] Z. Zhong *et al.*, "Access-time minimization in the IEEE 1687 network using broadcast and hardware parallelism," in *ITC'18*, 2018.
- [37] Z. Zhong *et al.*, "Broadcast-based minimization of the overall access time for the IEEE 1687 network," in *VTS'18*, 2018.
- [38] Z. Zhong, T.-C. Liang, and K. Chakrabarty, "IJTAG-based fault recovery and robust microelectrode-cell design for MEDA biochips," *TCAD*, 2020.
- [39] F. G. Zadegan *et al.*, "Design automation for IEEE P1687," in *Design, Automation & Test in Europe (DATE)*, 2011.
- [40] F. G. Zadegan *et al.*, "Test time analysis for IEEE P1687," in *IEEE Asian Test Symposium (ATS)*, pp. 455–460, 2010.
- [41] Z. Li *et al.*, "Error recovery in a micro-electrode-dot-array digital microfluidic biochip," in *International Conference on Computer-Aided Design (ICCAD)*, 2016.
- [42] Z. Li *et al.*, "Built-in self-test for micro-electrode-dot-array digital microfluidic biochips," in *IEEE International Test Conference (ITC)*, 2016.
- [43] Z. Li *et al.*, "Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips," *IEEE Transactions on Biomedical Circuits and Systems (TBioCAS)*, vol. 11, no. 3, pp. 612–626, 2017.
- [44] Z. Li *et al.*, "Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 37, no. 3, pp. 601–614, 2017.
- [45] Z. Zhong *et al.*, "Micro-Electro-Dot-Array digital microfluidic biochips: Technology, design automation, and test techniques," *TBioCAS*, vol. 13, no. 2, pp. 292–313, 2018.
- [46] Z. Zhong, T.-C. Liang, and K. Chakrabarty, "Reliability-oriented high-level synthesis for MEDA biochips," in *Asia and South Pacific Design Automation Conference*, 2020.
- [47] L.-P. Chang and C.-D. Du, "Design and implementation of an efficient wear-leveling algorithm for solid-state-disk microcontrollers," *ACM Transactions on Design Automation of Electronic Systems*, vol. 15, no. 1, pp. 1–36, 2009.
- [48] "Wear leveling for SSDs." <https://www.transcend-info.com/Embedded/Essay-22>, 2020 [online].
- [49] Y.-H. Chang, J.-W. Hsieh, and T.-W. Kuo, "Endurance enhancement of flash-memory storage, systems: An efficient static wear leveling design," in *ACM/IEEE Design Automation Conference*, pp. 212–217, 2007.
- [50] M.-C. Yang, Y.-H. Chang, C.-W. Tsao, and P.-C. Huang, "New ERA: New efficient reliability-aware wear leveling for endurance enhancement of flash storage devices," in *ACM/EDAC/IEEE Design Automation Conference*, pp. 1–6, 2013.
- [51] F. Su *et al.*, "High-level synthesis of digital microfluidic biochips," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 3, no. 4, pp. 1–32, 2008.
- [52] K. O'Neal *et al.*, "Force-directed list scheduling for digital microfluidic biochips," in *International Conference on VLSI and System-on-Chip (VLSI-SoC)*, pp. 7–11, 2012.
- [53] K. O'Neal *et al.*, "Resource-constrained scheduling for digital microfluidic biochips," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 1, pp. 1–26, 2017.
- [54] Z. Chen *et al.*, "Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture," *BioChip Journal*, vol. 5, no. 4, pp. 343–352, 2011.
- [55] M. Elfarg *et al.*, "Synthesis of error-recovery protocols for Micro-Electro-Dot-Array digital microfluidic biochips," *TECS*, vol. 16, no. 5s, pp. 127–148, 2017.
- [56] J. Hong *et al.*, "Three-dimensional digital microfluidic manipulation of droplets in oil medium," *Scientific Reports*, vol. 5, p. 10685, 2015.
- [57] R. S. Sista *et al.*, "Digital microfluidic platform to maximize diagnostic tests with low sample volumes from newborns and pediatric patients," *Diagnostics*, vol. 10, no. 1, p. 21, 2020.
- [58] S. Huang *et al.*, "Digital microfluidics for the detection of selected inorganic ions in aerosols," *Sensors*, vol. 20, no. 5, p. 1281, 2020.
- [59] D. Millington *et al.*, "Digital microfluidics comes of age: high-throughput screening to bedside diagnostic testing for genetic disorders in newborns," *Expert Review of Molecular Diagnostics*, vol. 18, no. 8, pp. 701–712, 2018.



Zhanwei Zhong received the bachelor's degree in electronic science and technology from Sun Yat-sen University, Guangzhou, China in 2013, the master's degree in electronic engineering from Tsinghua University, Beijing, China, in 2016, and the Ph.D. degree in computer engineering from Duke University, Durham, USA, in 2020.

He is currently a senior DFT engineer with Marvell Semiconductor Inc. in Santa Clara, CA, USA. He was a DFT intern with AMD Inc. (Beijing) in 2017 and with Apple Inc. (Cupertino) in 2019. His research interests include design automation for biochips, IJTAG test standard and design verification methodologies.



Tung-Che Liang received his B.S. degree in Electronics Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2014, and the M.S. degree from the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, in 2020, where he is currently working toward the Ph.D. degree.

He was with Synopsys Inc., Hsinchu, Taiwan, as an R&D engineer. He was a yield & diagnosis intern at Intel, Santa Clara, CA, a DFT intern at NVIDIA Inc., Santa Clara, CA, and a DFT intern at Apple Inc., Cupertino, CA. His current research interests include deep reinforcement learning, design automation, and security for microfluidic systems.



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 and Department Chair of Electrical and Computer Engineering and Professor of Computer Science at Duke University. 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 (2017), 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: design-for-testability of integrated circuits and systems (especially 3D integration and system-on-chip); microfluidic biochips; hardware security; neuromorphic computing systems. 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 received the 2018 Invitational Fellowship of the Japan Society for the Promotion of Science (JSPS) in the Short Term S (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.