Reliability-Oriented IEEE Std. 1687 Network Design and Block-Aware High-Level Synthesis for MEDA Biochips*

Zhanwei Zhong, Tung-Che Liang and Krishnendu Chakrabarty
ECE Department, Duke University, Durham, NC, USA
{zz114, tung.che.liang, krish}@duke.edu

Abstract—A digital microfluidic biochip (DMFB) enables miniaturization of immunoassays, point-of-care clinical diagnostics, DNA sequencing, and other laboratory procedures in biochemistry. A recent generation of biochips uses a microelectrode-dot-array (MEDA) architecture, which provides fine-grained control of droplets and seamlessly integrates microelectronics and microfluidics using CMOS technology. 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 present the design of an IEEE Std. 1687 (IJTAG) network and a block-aware high-level synthesis method that can effectively alleviate reliability problems in MEDA biochips. A comprehensive set of simulation results demonstrate the effectiveness of the proposed method.

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, commercially available DMFBs today 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 activate 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]. To ensure efficient execution of bioassays on MEDA biochips, a high-level synthesis tool is used to bind bioassay operations to on-chip resources, generate an optimized schedule of fluid operations, and determine droplet transportation routes [12]. It has also been reported in [13] that fluidic operations can fail during the execution of a bioassay due to biochip defects. These defects, which include damage to the hydrophobic layer, dielectric breakdown, and parasitic leakage, are often caused by excessive microelectrode usage [14]. It has been reported that erroneous bioassay outcomes may result in misleading prescription in point-of-care diagnostics, and they may threaten patients' lives [15]–[18]. Therefore, it is necessary to take reliability into consideration when we design MEDA biochips and generate high-level synthesis results for bioassays.

In order to alleviate the reliability problems for MEDA, we present a reliability-oriented high-level synthesis method. The key contributions of this paper are as follows:

- We analyze MEDA biochips and identify reliability issues associated with today's designs.
- We present IEEE Std. 1687 (IJTAG) network that divides the MEDA biochip into multiple blocks, and enables individual control of each block. Based on this design, *partial activation* and *partial sensing* procedures are proposed to reduce the number of data-shift operations and the overall usage of microelectrodes.
- We develop a block-aware high-level synthesis method that is specifically optimized for the proposed IJTAG network design. In this method, module placement and droplet routing are optimized to minimize the number of blocks being used. As a result, the number of data-shift operations and the overall usage of microelectrodes are further reduced.

The remainder of this paper is organized as follows. Section II describes the basics of microelectrode cells (MCs) in MEDA and the IJTAG network, and analyzes MEDA-specific reliability issues. Section III presents the IJTAG network design and the partial activation and partial sensing procedures. Section IV describes the block-aware high-level synthesis method that is optimized for the proposed IJTAG network design. Section V presents simulation results to illustrate the effectiveness of the proposed method. Finally, Section VI concludes the paper.

II. Background

A. Microelectrode Cell and Scan Chain

As shown in Fig. 1, 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 activation 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 activated to form a fluidic module (e.g., a splitter). When an MC is activated, 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 activated MC [9]. Otherwise, the bottom plate is floating

*This research was supported in part by the National Science Foundation under grant number CCF-1702596 and grant number ECCS-1914796.

978-1-7281-4123-7/20/$31.00 © 2020 IEEE

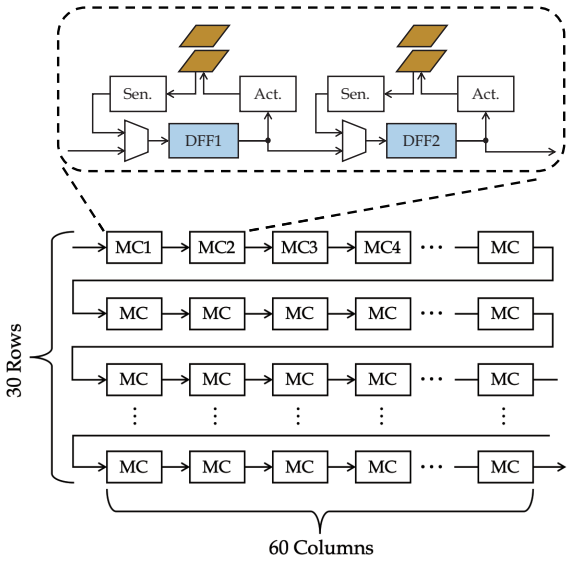

Fig. 1. The logic abstraction of MCs and the scan-chain structure in a fabricated 30 × 60 MEDA (adapted from [19]).

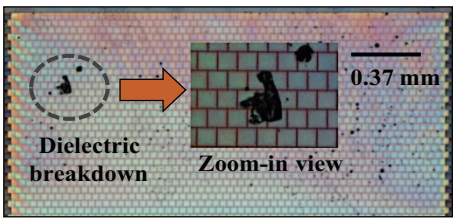

Fig. 2. Illustration of dielectric breakdown in a MEDA biochip [13].

and an induced voltage of 17 V is generated. In this case, no drag force is generated. The 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*. 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 *activation pattern*.

After MC-activation, 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; 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*. In the following discussion, we refer to the sensing value for each MC as the *MC-sensed value*. Note that when an MC-sensed value is latched in a DFF, it overwrites the MC-activation value. However, the newly-latched MC-sensed value will not change the status of the activation circuit, because in the MC-sensing process, all activation circuits are turned off.

B. Reliability Concerns

In MEDA biochips, the first reliability concern is **microelectrode degradation**. According to [14], when an electrode is excessively activated, charges will be trapped in the dielectric insulator, which can cause permanent electrode degradation. If this occurs, the force on a droplet is decreased, and droplets are likely to get stuck on the charge-trapped electrodes. In some extreme situations, excessive electrode activation can also lead to dielectric breakdown.

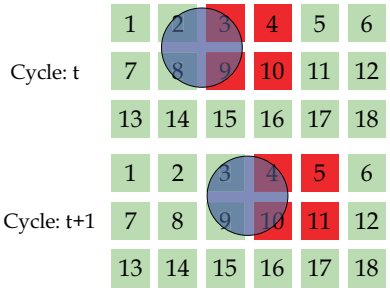

Fig. 3. The MC-activation status for two adjacent clock cycles.

Although the above issues have been reported for traditional DMFB platforms, MEDA biochips also suffer from these problem because a microelectrode in MEDA and an electrode in a traditional DMFB utilize the same working principles. One of these problems has already been reported for a fabricated MEDA biochip [13] (see Fig. 2). When droplet sensing is performed in an MC, the bottom plate of the MC is first charged to 3.3 V, and then discharged to 0 V. This process is repeated every second, which can easily cause charge trapping and dielectric breakdown.

The second reliability concern arises due to **excessive data-shift operations**. In a fabricated MEDA biochip with N_{mc} MCs, a total of $2N_{mc}$ bits (N_{mc} bits for MC activation/sensing) are shifted at a speed of 1 MHz each second. The power consumption for a MEDA biochip with N_{mc} MCs is given by $PC = (\frac{1}{2}C_{clk}V^2 + IP_{clk}) \times N_{mc} \times f + (\frac{1}{2}C_qV^2 + IP_d) \times N_{mc} \times f \times \alpha$ where C_{clk} is the input capacitance of the CLK pin in a DFF, V is the supply-voltage (3.3V for MEDA), IP_{clk} is the internal power consumption for each CLK signal switch, and f is the shift clock frequency (1 MHz for MEDA), C_q is the load capacitance of the Q pin in a DFF, IP_d is the internal power consumption for each D signal switch, and α is the activity factor [20]. The first term represents the power consumption to drive the CLK pins of DFFs. The second term represents the power consumption to drive the Q pins of DFFs.

According to the datasheet of the 0.35 μm process [21] and the detailed circuitry of an MC [22], the values of the parameters are: $C_{clk} = 0.013$ pF, $IP_{clk} = 0.4$ pJ, $C_q = 0.06$ pF, $IP_d = 0.5$ pJ. We have run simulations on a 60 × 30 MEDA biochip and a 300 × 150 MEDA biochip (25X in size), and the average activity factors are 0.12 and 0.04, respectively. Using these values, we note that the instantaneous power consumption for a 60×30 MEDA biochip and a 300×150 MEDA biochip are 1.02 mW and 22.6 mW, respectively.

Instantaneous power consumption may lead to increased noise, IR-drop and ground bounce issues, and it may also result in early-life failures [23]. From the above results, we can see that even if instantaneous power is not an issue for today's MEDA design with only 1800 MCs, it is likely to be a problem for larger biochips because the power consumption increases linearly with the number of MCs in MEDA.

We alleviate the above problems based on two key observations: (1) In each MC-activation step, only a small number of MC-activation values need to be updated; (2) In each MC-sensing step, only a small number of MCs need to perform droplet-sensing operations.

An example is shown in Fig. 3. Suppose at clock cycle t, a droplet that resides on the set of MCs 2, 3, 8, 9 need to move to the right. To do that, the MC sets of 3, 4, 9, 10 and 4, 5, 10, 11 are activated at clock cycles t and $t+1$, respectively. Because each MC stores the MC-activation value at cycle t in the DFF, to obtain the activation status for MCs at cycle $t+1$, we only need to shift in "0" values to the set 3, 9 and

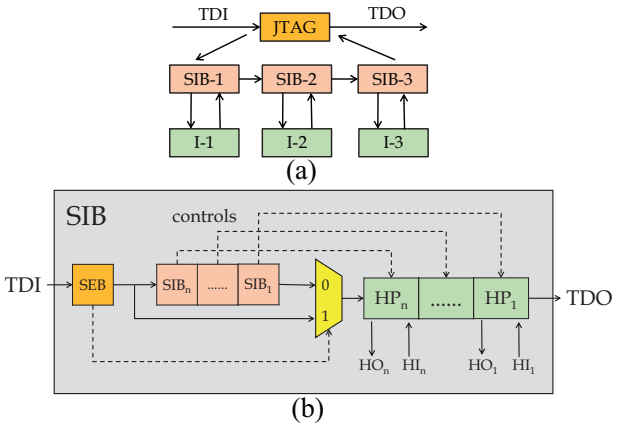

Fig. 4. (a) An example of an IJTAG network and (b) a simplified view of the SIB design.

"1" values to the set 5, 11. We can also see that at cycle t, the droplet resides on the set of MCs 2, 3, 8, 9 and the MCs 3, 4, 9, 10 are activated. In this case, we only need to perform droplet-sensing operations in MCs 2, 3, 4, 8, 9, 10 at cycle t, and then shift out the MC-sensed values.

The above two operations can significantly reduce the number of data-shift operations and the number MCs that perform droplet-sensing operations. However, these two operations cannot be implemented in today's MEDA design, because all MCs are serially connected in a single scan chain, and we cannot bypass some of the MCs for data-shift operations. Therefore, we need a new scan-chain design that can provide individual access to selected MCs, and this goal can be achieved using the IEEE Std. 1687 (a.k.a. IJTAG).

C. The IJTAG Network

The IEEE Std. 1687 (IJTAG) [24] 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 [25]–[27]. To provide flexibility of instrument access, a hardware component called the Segment Insertion Bit (SIB) has been introduced [28]. 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 [28].

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 includes 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 [29]. When SEB = 0, the SIB components are included in the primary scan path.

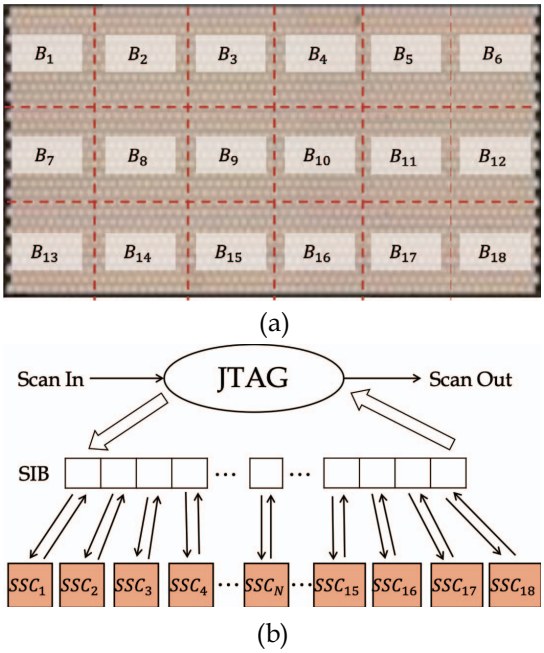

Fig. 5. (a) The floorplan for a 30×60 biochip with 10×10 blocks, and (b) the IJTAG network that is used to manage these blocks.

III. Proposed Hardware Design

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

A. IJTAG Network

In the new design, we first divide the biochip into multiple blocks. For a fabricated 30×60 MEDA biochip [9], if we define each block as a 10×10 rectangle, then the whole biochip is divided into 18 blocks (from B_1 to B_{18} shown in Fig. 5(a)). Next, the MCs in each block are sequentially connected to form a sub-scan chain (SSC). The SSC that corresponds to block B_i is referred to as SSC_i. Since there are 18 blocks in the biochip, the selected/unselected status of 18 SSCs are controlled by a 18-bit SIB register (see Fig. 5(b)). Note that the SIB register also provides a "sensing enable" signal to each block. If the ith bit of SIB register is "1", the sensing functionality of block B_i is enabled. Otherwise, it is disabled.

B. Partial Activation and Partial Sensing

In today's MEDA design, an actuation pattern is shifted in to update the MC-activation value for each MC. However, it is unnecessary because, as discussed in Section II, only a few MC-activation values need to be updated from cycle t to cycle $t+1$, and we only need to shift in the new activation values to MCs that need to be updated. Based on the new MEDA design, we first identify which MCs need to have new MC-activation values, and then determine which SSCs are involved. Next, we use the IJTAG network to select the corresponding SSCs, and shift in the new values to these SSCs. This operation is referred to as **partial activation**.

In addition to MC activation, in today's MEDA design, droplet-sensing operation is performed in all MCs to obtain the locations of droplets. However, as discussed in Section II, it is unnecessary to perform droplet sensing in all MCs. Instead, we only need to do it in the MCs on which droplet resides and the MCs that are activated in the current cycle. Based on the new MEDA design, we first identify which MCs need to perform droplet-sensing operations, and then determine which SSCs

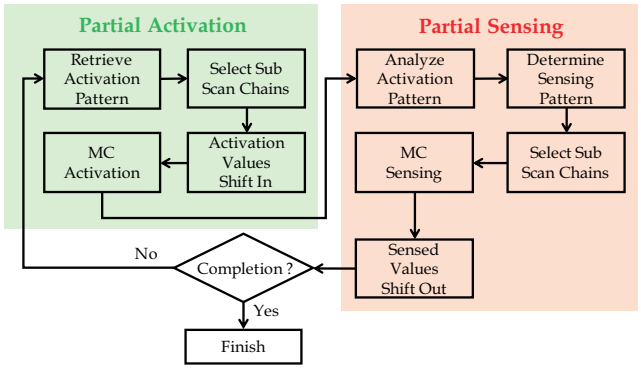

Fig. 6. The control flow for the new MEDA design.

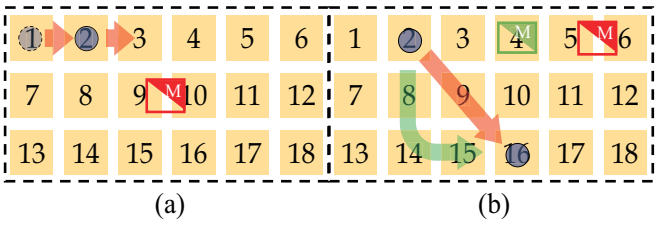

Fig. 7. Illustrative examples for (a) partial activation and partial sensing, and (b) block-aware module placement and droplet routing.

are involved. Next, we use the IJTAG network to select the corresponding SSCs, and enable the sensing functionalities. Finally, droplet sensing operations are performed in the MCs based on the selected SSCs, and the sensed results are shifted out from selected SSCs. This operation is referred to as **partial sensing**.

When an MC performs a droplet-sensing operation, the 0/1 MC-sensed value is written to the DFF, i.e., the MC-activation value is erased. However, the proposed partial-activation operation is based on the assumption that MC-activation values is preserved in the MCs. In order to preserve the MC-activation values, when MC-sensing values are shifted out, the erased MC-activation values are shifted in at the same time. Based on the partial activation/sensing operation described above, the control flow for the new MEDA design is shown in Fig. 6.

C. An Illustrative Example

In Fig. 7(a), each yellow rectangle represents a block, and there are 18 blocks in the biochip. Suppose at cycle t, a droplet resides on B_1, and all the MCs in B_2 are activated to drag the droplet from B_1 to B_2. At the same time, a fluidic module resides on both B_9 and B_{10}. We also assume that at time $t+1$, the droplet moves to B_2, and all the MCs in B_3 are now activated to drag the droplet from B_2 to B_3. The fluidic module is still at the same position. In this case, partial activation and partial sensing operations are performed at cycle $t+1$ as follows:

Partial Activation: Based on the MC-activation values at cycle t, we only need to update the activation-values for the MCs in B_2 and B_3. We first configure the IJTAG network, and select SSC_2 and SSC_3. Next, "0" are shifted in B_2 and "1"s are shifted in B_3.

Partial Sensing: Based on the activated MCs and the droplet locations at cycle $t+1$, we can determine that only the MCs in B_2, B_3, B_9 and B_{10} need to perform droplet-sensing operations at cycle $t+1$. We first configure the IJTAG network, and select SSC_2, SSC_3, SSC_9 and SSC_{10}. Next, droplet sensing is performed in the MCs corresponding to these four SSCs, and the MC-sensed values are then shifted out.

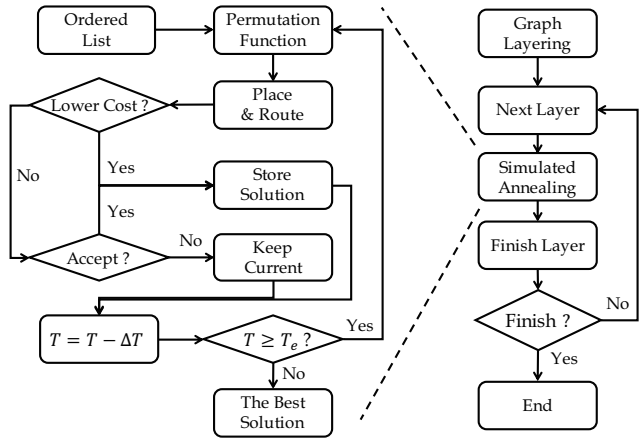

Fig. 8. The overall flow for the high-level synthesis designed for partial activation and parting sensing.

In cycle $t+1$, only a total of six SSCs are used for data-shift operations (two for partial activation and four for partial sensing), which is a large reduction (83%) compared with the 36 SSCs in previous MEDA designs. In addition, the MCs of only four SSCs perform droplet-sensing operations, which is also a large reduction (77%) compared with the MCs of 18 SSCs in previous MEDA designs.

IV. Block-Aware High-Level Synthesis

A. Motivation and Problem Formulation

Fig. 7(b) shows why optimized high-level synthesis is needed. Each yellow rectangle represents a block, and there are 18 blocks in the biochip. Suppose a droplet needs to move from B_2 to B_{16}. A fast and straightforward way to attain this goal is to move the droplet in the diagonal direction (red arrow). However, in this process, the MCs in B_2, B_3, B_8, B_9, B_{10}, B_{15} and B_{16} need to be activated to perform droplet-sensing operations. However, if we move the droplet along the block boundary (green arrow), only the MCs in B_2, B_8, B_{14}, B_{15} and B_{16} are used, which is 28% less than for the diagonal route.

A discussion of module placement is also relevant; see Fig. 7(b) again. If a fluidic module is placed on both B_5 and B_6 (indicated by the red box), the MCs in B_5 and B_6 are used for partial activation and partial sensing. However, if we move this fluidic module to B_4 (indicated by the green box), then only the MCs in B_4 are used, which is 50% less than the number of MCs needed for the previous module placement.

From these two examples, it can be concluded the high-level synthesis algorithm should consider the blocks to determine module placement and droplet routing. We therefore present the following problem formulation:

Input: (1) The sequencing graph $G = \{V, E\}$ for the bioassay, where V represents a fluidic module and E represents the dependencies between all pairs of fluidic modules [22]; (2) The MEDA biochip library, which includes the type, size and corresponding execution time of on-chip fluidic functional modules; (3) The size of the MEDA biochip; (4) The IJTAG network design for MEDA biochips.

Output: Schedule of operations, module placements and droplet routing.

Objective: Minimize the cost function C given by $C = \sum_{t=1}^{T}(N_{pa}(t)+N_{ps}(t))$ where T is the number of clock cycles needed to complete the bioassay, $N_{pa}(t)$ and $N_{ps}(t)$ are the numbers of blocks that need to be used for partial activation and partial sensing operations at clock cycle t, respectively.

B. The Overall Flow

The overall flow of the high-level synthesis method is shown in Fig. 8. It includes two loops: an outer loop and an inner loop. In a sequencing graph, the weight on a vertex denotes the time used by a fluidic module, and the weight on an edge denotes the time used by the droplet route between fluidic modules. In the outer loop, we first schedule the fluidic modules using the As-Late-As-Possible (ALAP) algorithm. Note that the droplet routing times are still unknown, therefore the weights of the edges are set to "0"s.

In the schedule, each vertex is assigned a *start time*, and the sequencing graph is layered according to vertex start time. Next, we use a sliding window to include the first and second layers, and form a two-layer subgraph. In this subgraph, the first layer is defined as the *parent layer*, and the second layer is defined as the *child layer*. Because vertices in the parent layer are dispensing modules and their locations are pre-defined, we only need to determine the module placement for the vertices in the child layer and also droplet routes from the dispensing ports in the parent layer to vertices in the child layer. This goal is achieved by using the simulated annealing (SA) technique. After we determine the module placement and droplet routing for the current sub-graph, we move the sliding window down one layer and form a new two-layer sub-graph. This process repeats until the module placement and droplet routing for vertices from the bottom layer of the sequencing graph are completed.

In simulated annealing, we first set the initial and ending temperature to user-defined values T_i and T_e, respectively. Next, an initial ordered list of child vertices is generated, and a permutation function is used to randomly switch the positions of two vertices. For example, suppose we have four child vertices A, B, C and D in the subgraph. If the initial ordered list is [A, B, C, D], then the permutation function can change the ordered list to [A, D, C, B] with the positions of vertices B and D being switched. After we obtained a permuted ordered list, a *place_and_route* algorithm (see Section IV.C) generates optimal module placement and droplet routes for the child vertices in the permuted ordered list, and we compute the following cost $C(sg) = \sum_{t=1}^{T(sg)}(N_{pa}(sg,t) + N_{ps}(sg,t))$ where $T(sg)$ is the number of clock cycles to complete the module placement and droplet routes for all child vertices in the current subgraph, and $N_{pa}(sg,t)$ and $N_{ps}(sg,t)$ are the numbers of blocks that are used for partial activation and sensing operations at clock cycle t. Finally, we examine whether cost $C(sg)$ is the smallest for the current subgraph. If it is the case, the corresponding ordered list of child vertices is stored, and the next permutation function is operated based on this ordered list. Otherwise, an "Accept" function is used to determine whether we need to store this ordered list.

If the temperature T is high, there is a higher probability to "accept" and store this ordered list. Otherwise, it is more likely that the algorithm will discard this ordered list. The "Accept" function reduces the likelihood of the solution being stuck in local minima.

C. Place and Route for Sub-graph

The pseudocode for the $place\_and\_route$ algorithm is shown in Fig. 9. The input to this algorithm is the two-layer subgraph, and the ordered list of child vertices. For each vertex in the child layer, we first determine the minimum and maximum numbers of blocks the corresponding fluidic module can occupy (line 4). Next, we carry out a linear search on the number of block being occupied, namely num_b (line 5). For each value of num_b, we find out a set of module positions that occupies num_b of blocks (line 6) and select

Algorithm 1 place_and_route(subgraph, ordered_list, p)

Input: The subgraph, the ordered list, and the trade-off factor p;
Output: The placement and routing for the children layer.

```
1:  placement_set := {};
2:  route_set := {};
3:  for each vertex in current_layer do // according to vertex_order
4:      min_b, max_b := get_num_block(vertex);
5:      for num_b := min_b to max_b do
6:          location_set := find_location_set(num_b);
7:          while p <= 1 do
8:              best_location := find_best_location(location_set, p);
9:              route := droplet_router(best_location);
10:             if route != None then break;
11:             else p := p + 0.2;
12:         if route != None then
13:             placement_set[node] = best_location;
14:             route_set[node] = route; break;
15:         if (route = None) and (num_b = max_b) then
16:             exit; // place_and_route fails
17: return placement_set, route_set;
```

Fig. 9. The pseudocode for place_and_route algorithm.

the location that minimizes the following cost function $C(loc)$ (line 8): $C(loc) = (1-p) \times \sum_{p\in\mathbb{P}} Man(loc(p), loc) + p \times \sum_{c\in\mathbb{C}} olp(BB(loc(c)), BB(loc))$ where $\mathbb{P}$ is the set of predecessors for the child vertex, $Man(A,B)$ is the Manhattan distance from position A to position B, $loc(p)$ is the location of predecessor p, and loc is the location that is evaluated. A lower value of the first term indicates a shorter length for droplet routes and thus less blocks used.

In the second term, $\mathbb{C}$ is the set of child vertices in the current subgraph whose module placement and droplet routes are determined. $loc(c)$ is the location of vertex c, and $BB(loc(c))$ are the bounding boxes correspond to the net from vertex c's predecessors to the location of vertex c. $BB(loc)$ are the bounding boxes correspond to the nets from the current child vertex's predecessors to the location of the current child vertex. Here, a *bounding box* for a net is defined as the rectangle area of the maximum extent of the source pin and sink pin. $olp(BB_1, BB_2)$ is the sum of mutually overlapping area among bounding boxes BB_1 and BB_2.

Finally, p is a trade-off parameter. In the algorithm, p 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, the increase of parameter p can alleviate the problem of routing congestion and make droplet routing easier.

After we select the target location for a child vertex, we determine the routes that come from its predecessors (line 9). 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 clock cycles. We use the 3D A^*-algorithm [30] to obtain the shortest route from a parent vertex to a child vertex. In this case, the number of blocks used for partial activation can be kept small. This process repeats until we determine the locations and droplet routes for all child vertices.

However, if a route cannot be found from a parent vertex to the target location corresponding to a child vertex, we increase the value of p (line 11), find a new target location for the child vertex, and re-route is performed. This process is repeated until we successfully find a route from a parent vertex to the target location (line 12-16).

V. Experimental Results

We use three real-life benchmarks, namely CEP, serial dilution, and master mix [31] for evaluation. CEP is a combination of three small bioassays: cell lysis, mRNA extraction, and

TABLE I
SIMULATION RESULTS FOR THREE REAL-LIFE BIOASSAYS.

	Bioassay Protocols								
	CEP			Serial Dilution			Master Mix		
Method	CT	DS	CS	CT	DS	CS	CT	DS	CS
M-1	51	183.6K	91.8K	38	136.8K	68.4K	81	291.6K	291.6K
M-2	51	30.1K	16.1K	38	55.4K	28.5K	81	62.2K	32.2K
M-3	54	17.3K	9.5K	40	42.2K	22.8K	90	44.9K	23.5K

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 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 [22].

We use the following three evaluation metrics: (1) The bioassay completion time (**CT**) (unit: s), (2) the **total** number of data-shift operations (**DS**), and (3) the **total** number of MCs that perform droplet-sensing operations (**CS**). The metrics of **DS** and **CS** are: $DS = \sum_{t=1}^{CT}(N_{pa}(t) + N_{ps}(t)) \times N_{mc}$ and $CS = \sum_{t=1}^{CT} N_{ps}(t) \times N_{mc}$ where $N_{pa}(t)$ and $N_{ps}(t)$ are the number of blocks used for partial activation and partial sensing operations at clock cycle t, respectively. The parameter of N_{mc} refers to the number of MCs in each block.

In order to quantify the benefits introduced by the IJTAG network (partial activation and partial sensing) and the block-aware high-level synthesis, we compare the following three methods:

M-1: The high-level synthesis method in [22] for the baseline MEDA design.

M-2: The high-level synthesis method in [22] applied to the new MEDA design with the new idea of partial activation/sensing operations being utilized.

M-3: The proposed block-aware high-level synthesis method applied to the new MEDA design, and partial activation/sensing operations are utilized.

The simulation results are shown in Table I. For M-1, the values of **DS** and **CS** are large because all MCs are used for data-shift and droplet-sensing operations. However, if partial activation/sensing operation is used (i.e., in M-2), we can achieve up to 80% reduction in **DS** and **CS** because only the MCs that are needed are used in partial activation/sensing operations. Finally, when block-aware high-level synthesis is used (i.e., in M-3), we can see that, compared with the results for M-2, an additional 50% reduction is achieved for **DS** and **CS**. This benefit comes at the cost of slightly longer completion times for the three bioassays. However, because M-3 is mainly used to reduce the values of **DS** and **CS**, and it is not optimized for bioassay completion time, the completion times for M-3 are slightly larger than those for M-1 and M-2 for the three benchmarks.

VI. CONCLUSION

In this paper, we have first highlighted some key reliability challenges associated with today's MEDA biochip designs. Next, we have presented an IJTAG network design and a corresponding new control flow, which can significantly reduce the number of data-shift operations and the number of MCs that performs droplet-sensing operations. Subsequently, we have introduced a block-aware high-level synthesis method tailored for the new MEDA design. 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] 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.

[13] 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, 2017.

[14] 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.

[15] 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.

[16] T.-C. Liang *et al.*, "Execution of provably secure assays on meda biochips to thwart attacks," in *ASPDAC'19*, pp. 51–57, 2019.

[17] 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.

[18] 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.

[19] L. Zhang *et al.*, "Built-in self-diagnosis and fault-tolerant daisy-chain design in MEDA biochips," in *ITC'18*, 2018.

[20] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.

[21] "AMI 0.35 μm PDK." https://vlsiarch.ecen.okstate.edu/flow, 2019.

[22] Z. Li *et al.*, "High-level synthesis for micro-electrode-dot-array digital microfluidic biochips," in *DAC'16*, 2016.

[23] C. F. Hawkins *et al.*, "Test and reliability: Partners in IC manufacturing, part 1," *IEEE Design & Test of Computers*, no. 3, pp. 64–71.

[24] IEEE Standard Committee *et al.*, "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Standard*, 2014.

[25] R. Cantoro *et al.*, "Automatic generation of stimuli for fault diagnosis in IEEE 1687 networks," in *IOLTS'16*, pp. 167–172, 2016.

[26] Z. Zhong *et al.*, "Access-time minimization in the IEEE 1687 network using broadcast and hardware parallelism," in *ITC'18*, 2018.

[27] Z. Zhong *et al.*, "Broadcast-based minimization of the overall access time for the IEEE 1687 network," in *VTS'18*, 2018.

[28] F. G. Zadegan *et al.*, "Design automation for IEEE P1687," in *DATE'11*, pp. 1–6, 2011.

[29] F. G. Zadegan *et al.*, "Test time analysis for IEEE P1687," in *ATS'10*, pp. 455–460, 2010.

[30] 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.

[31] M. Elfar *et al.*, "Synthesis of error-recovery protocols for Micro-Electrode-Dot-Array digital microfluidic biochips," *TECS*, vol. 16, no. 5s, pp. 127–148, 2017.