Journal of Electronic Testing (2023) 39:521–534
https://doi.org/10.1007/s10836-023-06081-8

Online Diagnosis and Self-Recovery of Faulty Cells in Daisy-Chained MEDA Biochips Using Functional Actuation Patterns

Ling Zhang[1]

Received: 5 April 2023 / Accepted: 23 August 2023 / Published online: 16 September 2023

Abstract
Digital microfluidic biochips are becoming an alternative for laboratory experiments like DNA analysis, immunoassays and safety clinical diagnostics. Reliability is a critical issue for them. MEDA biochips are a new kind of digital microfluidic biochips which are based on microelectrode-dot-array. Daisy chain is the key component of a MEDA biochip, if there is one fault exist in one cell, the whole daisy chain is broken down, the biochip has to be discarded. Therefore, daisy chain's reliability is especially critical. The paper proposes a new daisy chain online repair scheme, where the cells of daisy chain are repaired during the time of the functional data shifted in. The emphasis is on identifying and repairing the fail daisy chain cells when bioassay is running, without any influence on the executing bioassay. Besides, the online self-repair scheme does not contribute to electrode degradation, and takes no additional time. The experimental results also prove the efficiency of the scheme.

Keywords MEDA biochip · Reliability · Daisy chain · On-line repair · Fault tolerant

1 Introduction

A Digital microfluidic biochips (DMFB) manipulates liquids as nanoliter or picoliter droplets based on the principle of electrowetting-on-dielectric. It is a promising platform and has become widely used on high-throughput DNA sequencing [21], point-of-care clinical diagnostics [3], and protein crystallization in drug discovery [10]. DMFBs offer the advantages of simple instrumentation and flexible device geometry and are easy to be reconfigured and integrate with other technologies [4, 6, 8, 9, 29]. There are a number of startups working on transitioning digital microfluidic technology to commercialization: miroculus worked on developing a molecular-testing solution using an automated digital microfluidic platform [18]. Illumina, has utilized digital microfluidic technology for library preparation automation and high-throughput genomics [11]. Baebies also developed a digital microfluidic-based platform for disease screening of newborns and COVID-19 detection [1]. These milestones highlight the emergence of DMFB technology for commercial use.

Recently, a new kind of DMFB based on a microelectrode-dot-array (MEDA) architecture has been proposed [7, 12]. Compared with conventional DMFBs, different droplet sizes can be manipulated in a fine-grained manner for MEDA biochips. The architecture allows microelectrodes to be dynamically grouped to form a micro-component that can perform various microfluidic operations on the chip. The basic unit of a MEDA biochip is a microelectrode cell (MC) composed of a microelectrode and the associated actuate/sense circuits. So the MEDA biochips can incorporate real-time capacitive sensing on every microelectrode to detect the properties and locations of droplets. The "sensing map" derived in this manner presents an exciting opportunity for cyber-physical MEDA biochips that can dynamically respond to bioassay outcomes, perform real-time error recovery, and execute "if-then-else" protocols from biochemistry. Prototypes of MEDA biochips have been fabricated using TSMC 0.35 μm CMOS technology [12], and these devices can use a power-supply voltage of only 3.3 V [7].

MEDA biochips integrate additional components such as actuate/sense circuits. Like the case of integrated circuits, the increase in the density and the area of microfluidic biochips can further lead to increased defects and

Responsible Editor: K. Chakrabarty

✉ Ling Zhang
zhangling@jxufe.edu.cn

[1] School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330003, JiangXi, China

decreased yield [13, 15, 17]. Defects can lead to errors that adversely affect the accuracy of the entire bioassay experiment [14, 36]. Reliability is a critical issue for biochips, especially for safety-critical applications. So MEDA biochips not only need to be tested before being used, but they also need to be tested online. Furthermore, they should be designed to tolerate faults and operate reliably even when defects occur.

To solve the above problem, we propose a more specific and flexible approach that targets on test, diagnosis, and repair of the daisy chain automatically in the running time of the bioassay. The key contributions of this paper are as follows.

1. Analyzes work principles and reliability of a MEDA biochip. Gives the summary that the daisy chain's reliability is critical for a MEDA biochip.
2. presents the self-repair module and a daisy chain diagnosis scheme that works in a polling manner, with MCs diagnosed and repaired serially.
3. Analyzes work features of an MC with a self-repair module and proposes a new repair scheme target on the defects that occur in the daisy chain in the bioassay running time. It takes the functional data as test vectors, has no contribution to the electrode degradation and takes no additional time.
4. Through analysis of the actuation patterns, we found that the actuation patterns are very sparse. Accordingly, the don't care bits of spare electrodes in the actuation patterns are used to generate optimized actuation patterns/test vectors, which can repair as many faults as possible. Besides, we demonstrate the effectiveness of the proposed scheme through extensive experiments.

The remainder of this paper is organized as follows. In Section 2, the defect model and the related work to improve the reliability of the MEDA biochips are presented. The daisy chain structure and its reliability are also analyzed in this section, illustrating the motivation of this work. Section 3 gives the work principles of MC and then discusses the work features of the MC with a self-repair module, also proposing the new repair scheme using actuation patterns. The experimental results are given in Section 4. Finally, conclusions are drawn in the last section.

2 Background and Motivation

This section presents defects models, related works for improving reliability, and the reliability analysis for MEDA biochips.

2.1 The Defect Models of the MEDA Biochip

For MEDA biochips, defects can arise during the manufacturing and bioassay execution and may be caused by hardware or software failures. Reliability is essential for them, especially when used for point-of-care diagnostics and health assessments [36]. Therefore, the MEDA biochip should be adequately tested before and during bioassay execution to find defects and malfunctions. Furthermore, the discovered defects or malfunctions during the bioassay execution should be tolerated.

The four typical hardware defects models related to the biochip components caused by manufacturing defects or device damage and aging are as follows [22, 25, 36]. (1) Hydrophobic Layer Defects: A standard hydrophobic layer can increase the electrowetting force when transporting droplets. The electrowetting force will weaken and be insufficient to correctly actuate the droplets when hydrophobic layer defects occur [3]. (2) Dielectric Breakdown: Applying high voltage to the electrode can cause electrode breakdown, resulting in the droplet's direct contact with high voltage, and the droplet will be electrolysis and uncontrollable. (3) Short-Circuited Microelectrodes: Adjacent electrodes are short-circuited to form "microelectrodes" that cannot be controlled independently, so the droplet passing through these microelectrodes may cause errors. (4) Transistor Failures: Transistors exist in control/sensing circuits or daisy chains, which are the daisy chain's primary components. Transistor failure can cause droplets to be driven, maintained or sensed incorrectly. Traditional scan chain fault models can be utilized to describe the defects.

Besides hardware defect models, the four typical malfunction fault models in MEDA biochips are Dispensing, Transportation, Mixing, and Split failure. Dispensing failure is caused by the irreversible charge concentration on the dispensing electrode, where droplets cannot be dispensed from the reservoir. Transportation failure leads to insufficient electrowetting force when the microelectrode is actuated. Mixing and Split failures lead to the malfunctions of mixing and splitting operations.

2.2 Related Work to Improve the Reliability of the MEDA

In order to improve the reliability of MEDA biochips, many works has been done to design further testing, diagnostic, repair or fault tolerance methods for different fault models of MEDA chips. Because a micro-sensor is embedded under each microcell of the MEDA chip, the "sensing map" can be checked in time on the controller side.

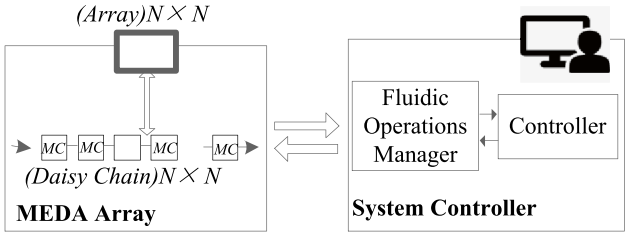


Fig. 1 Block diagram of a MEDA biochip

Therefore a considerable amount of work has been done to reduce the occurrence of malfunctions. On the contrary, little work specifically for hardware failures. Here we introduce the related works for the two failure categories to highlight the differences in the work of this article.

2.2.1 Reliability Work Target on Hardware Defects

The droplet-based offline methods control the test droplets stored in a reservoir to pass through the biochip array's cells to identify and diagnose the defects [24, 26, 27]. However, dispensing and manipulating test droplets takes significant time, preventing rapid testing. In addition, the external capacitive sensing circuits were used to analyze the test outcomes to track the motion of the test droplet, where errors are prone to occur because of inaccurate sensor calibration.

The built-in self-test method [17] designs the test pattern generators (TPGs) to offer test patterns and the response comparator (RC) to check the failure of the corresponding components under the control of the BIST controller. It provides a scan chain test, intro-MC test and inter-MC test. A similar work target on the scan chain and MC test is also proposed in [13], which is to verify the integrity of the scan chain and the droplet maintenance, actuation and sensing for each MC. An oscillation-based testing (OBT) [23] converts the test operations into a simple oscillator, where the fault-free circuits generate a standard frequency. In contrast, the deviated frequency is produced for the faulty circuit.

2.2.2 Reliability Work Target on Malfunctions

The functional testing is to verify the reliability of the biochip at the fluidic operation level. It targets different functional fluidic operations to make sure they can be used to perform certain functions correctly. e.g., dispensing, transportation, mixing, and splitting [30, 31]. The MEDA functional test is divided into splitting, mixing, dispensing and routing tests [13], where splitting and mixing tests can be done together for they are highly related.

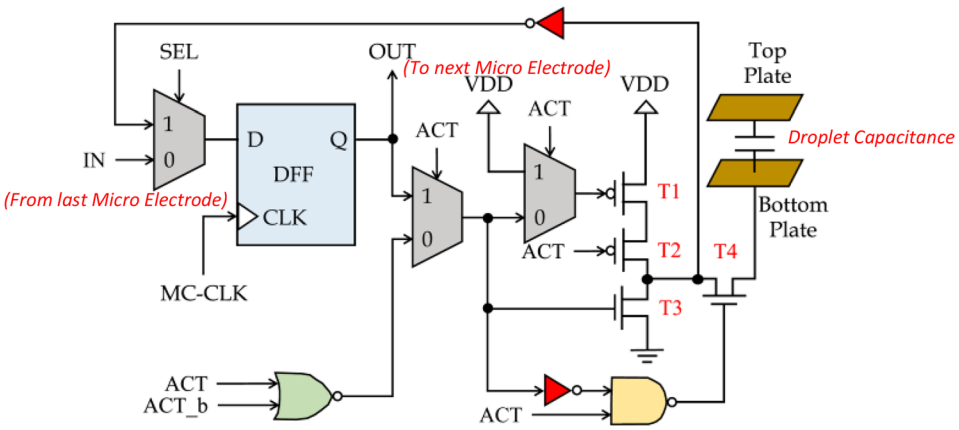
The error-recovery schemes are making error-recovery to ensure the accuracy of the bioassays. Generally, this kind of work uses the "sense map" to discover the malfunctions of executed bioassay in real-time and then use different error recovery approaches to ensure the reliability of the bioassay.

Paper [14] proposed an error-recovery technique that dynamically reconfigures biochips according to the real-time sensed data obtained from on-chip sensors. To fully utilize MEDA-specific capabilities, paper [40] employed the droplet-aliquot operation to enable error recovery after droplet splitting. The methods in [5, 37] formalized error-recovery objectives and synthesized optimal error-recovery protocols using a model based on stochastic multiplayer games. Moreover, fault-recovery solutions based on the homogeneous structure of MEDA were presented in [35, 38, 39]. These works mainly to make error recovery for MEDA biochips. They all suppose that the daisy chain and the sensors are reliable. If the daisy chain infrastructure is broken, most existing test and fault-tolerant methods cannot be used.

These test works can only be used offline before being sold to consumers or before being used. Moreover, they cannot repair the faulty cell. If there are any detected faults, the biochip has to be discarded.

The first built-in self-diagnosis and fault-tolerant daisy chain structure for MEDA biochips were proposed in paper [33] to improve the reliability of the online biochip and tolerate the defect defects. It can not only identify the faulty cell of the daisy chain but also can repair them automatically.

Fig. 2 Schematic of an MC [12]



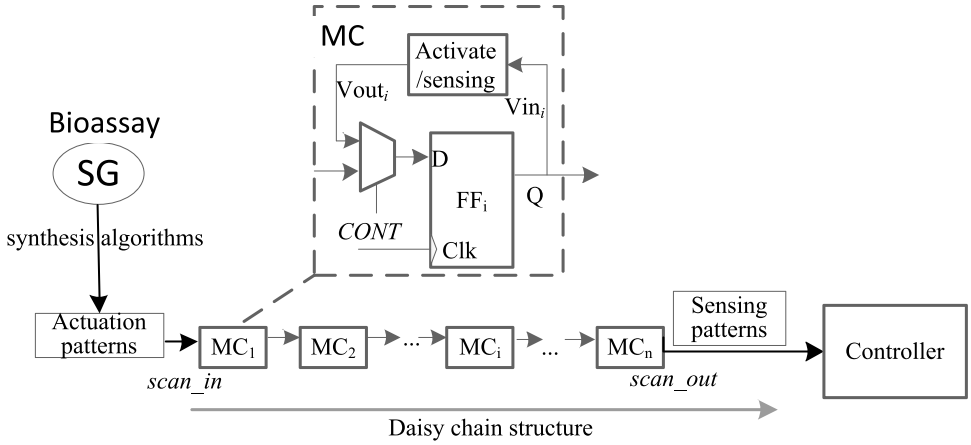


Fig. 3 The daisy chain structure and its use for bioassay execution

The enhanced online daisy chain fault-tolerant scheme proposed in [34] gives the detailed test, diagnosis and repair process for daisy chains of MEDA biochips. However, they still couldn't test and repair the failed cells of the daisy chain during the bioassay was running.

2.3 The Reliability Analysis for MEDA Biochip

The architecture diagram of a MEDA biochip is given in Fig. 1. The two main blocks are the System Controller and the MEDA biochip array. In the MEDA biochip array, MCs are daisy-chained together to simplify wiring. The system controller can be entirely or partially integrated into the biochip, depending on fabrication technologies and application requirements. The controller is typically a computer with sufficient memory space, interface circuitry, and software programmability [29]. It reserves network connectivity used for loading new bioassays.

An MC schematic diagram is shown in Fig. 2. Due to the high integration level of MEDA biochips, MCs are connected by a daisy chain, i.e., the output of one MC is connected to the input of the next MC. The daisy chain achieves complete control of every microelectrode with only seven control pins: *CONT*, *SET*, *RESET*, *ACT*, *ACT_b*, *IN*, and *OUT* [12].

The daisy chain structure and its use for bioassay execution are given in Fig. 3. For the sake of simplicity, only the register and the multiplexer in Fig. 2 are shown in Fig. 3; other elements are included in the Activate/Sensing Block. When a MEDA biochip works in functional mode, a single actuating bit is stored in the register of each MC. The single-bit determines whether the MC actuates or maintains the corresponding droplet. These single bits for all MCs in some same actuation cycles form an actuation pattern. An actuation pattern shifted serially through the daisy chain. A frame clock cycle is the time needed to shift an actuating pattern. In the sensing mode, the sensing control signal *CONT* selects the signal from the actuation/sense module. Then the sensed results are stored in the corresponding MCs registers, forming a sensing pattern. The sensing pattern is then serially shifted out through the daisy chain [33].

A bioassay is often expressed as a sequence graph (SG). Then a synthesis algorithm is used to convert the SG into a sequence of actuation patterns. The application of these actuation patterns completes the bioassay execution: (1) One actuation pattern is shifted into the daisy chain and arrives at the register of each MC. The daisy chain works in the actuation mode, where the actuation data in each register will actuate or maintain the corresponding liquid droplet according to the operation plan of the bioassay. (2) The daisy chain works in sensing mode/detection mode, and each MC performs droplet sensing (detection). The result is clocked into the shift register and becomes the 1-bit map data. Droplet-sensed results are serially shifted to the controller for readout, forming the droplet location map. (3) When the controller receives the location map, the microfluidic manager uses the location map to decide the next actuation control data. If not all droplets are in the desired locations, the last actuation pattern will be executed again. Otherwise, actuation control data of the following actuation pattern is serially loaded into the daisy chain. The time from the start of the sensing operation to the shifting completion of the next actuation pattern forms a Detection/Control window.

Compared with the time scale of the physical movement of the droplets (several hundred ms), the clock frequency of the daisy chain is very high, and the actuation and detection time are typically less than 10 ms. Thus the actuation data and the sensing data shifting time can be neglected. To meet the timing requirement for larger-scale DMFBs, higher frequency clocks can handle the loading and readout [28].

The daisy chain has the same structure as the scan chain in integrated circuits(ICs). A scan chain is easy to break down, and 50% of faults of ICs are caused by scan chain failure [32]. A bioassay's functional and sensing data must be transmitted from the daisy chain. Although the scale of the daisy chain is smaller than that of the integrated circuit, there is no guarantee that it will be error-free.

Generally, the test of a DMFB is performed in two phases: offline and online testing. In offline testing, the test techniques are applied to the biochips after manufacturing. If a biochip is tested simultaneously with the bioassay operation process, it is called online or concurrent testing [2].

The target of the offline is to make sure the reliability of biochips before it is sailed to the consumers. It will be drifted to the consumers if it has passed the test without any failure. Consumers use the biochip for commercial usage or experimental usage. Online testing is used to judge the biochip's correctness before it is used. If it has passed the test and has no fault, it will be used for running a bioassay. Otherwise, if there are some failed electronics, a re-synthesis algorithm will be applied to get the correct assay operations to avoid faulty MCs. The biochip must be discarded if there are too many failed electronics.

A test process of today's MEDA biochips is shown in Fig. 4. For manufacturers, offline testing is applied to check if the biochip is faultless before being sold to the consumers, and the scan chain is also tested. If the daisy chain has a defect, the biochip has to be discarded. For consumers, the biochip will be tested before each is used. The biochip must be discarded if the daisy chain has even one fault. Thus, the reliability of the daisy chain is very critical.

Though papers [33, 34] propose the schemes for repairing faulty cells of a daisy chain through additional test vectors. However, it did not consider the reliability of a running bioassay. For a bioassay executed on the MEDA biochip, if a defect occurs in the daisy chain during its running time, the biochip system would break down, resulting in the biochip's reduced service life and reagent waste.

2.4 The Target of this Paper

As presented above, the daisy chain is the backbone of the MEDA biochip. All running process of the MEDA biochip depends on the faultless daisy chain. It requires extremely high reliability. Therefore, efficient fault-tolerant strategies during the execution of the bioassay need to be designed to ensure robust fluidic operations and accurate bioassay outcomes.

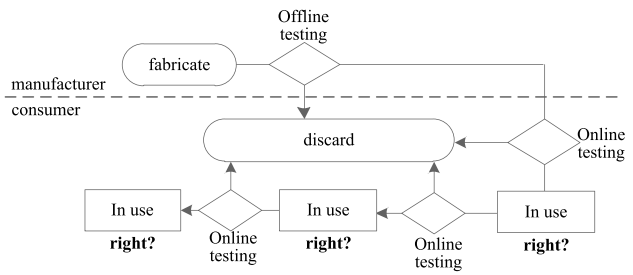


Fig. 4 The test diagram schematic of MEDA biochips

Like the scan chain in IC, the daisy chains are prone to fail. Though the scale is comparably smaller than scan chains of ICs, there is no guarantee that it will not fail. Transistor failures due to aging or physical damage during bioassay execution cannot be guaranteed either. Based on the analysis, this paper proposes an online test method for the daisy chain and a fault tolerance scheme for daisy chain transistors based on the daisy structure in article [34]. The target is to test, diagnose and repair the defects in the daisy chain cells during the execution time of the bioassay. It uses the actuate patterns of the bioassay to complete the test and fault tolerant process to ensure the online reliability of the bioassay with no additional time or hardware cost.

3 Online Diagnosis and Self-Recovery of Faulty Cells in Daisy-Chained MEDA Biochips Using Functional Actuation Patterns

Based on the self-repair daisy chain structure proposed in [34], this paper presents a method for online testing, diagnosing and repairing daisy chains using functional actuation patterns. This section will first introduce the daisy chain structure with self-repair functions. Then, the structure's features will be analyzed and combined with the bioassay workflow to design an optimal scheme for improving the daisy chain's reliability. Finally, the proposed self-repair method for a MEDA biochip's running process is described.

3.1 Daisy Chain Structure with Diagnose and Repair Function

Figure 5 gives the daisy chain structure with diagnose and repair functions [34]. The daisy chain with n MCs is given in the figure. These n MCs are visited serially under the control signals of *ctl*s. The *ctl*s are controlled in a one-hot manner. Only one *ctl* is set to logic-1 in a clock cycle. R is a fault indicator, where R=1 will invoke the repair of the MC.

The detailed work principle is described as follows. In the first cycle, ctl_1 is set to logic-1, and the other *ctl*s are set to logic-0. Only MC_1 is visited in this cycle. The data bit shifted from *shift_in* will be written in the 1-bit register of MC_1. It is also sent to the Controller simultaneously to check the correctness. If the captured bit is wrong, the indicating signal R will be set as 1, otherwise 0. In the second cycle, ctl_2 for MC_2 is set as logic-1, other *ctl*s are set as logic-0, and MC_2 works similarly. Then MC_2-MC_n are revisited. The control signals ctl_1-ctl_n can be easily generated internally by the control signal generator. No additional pins are required. The control signal generator can be implemented using multiple on-chip clock controllers, counters, and decoders [20].

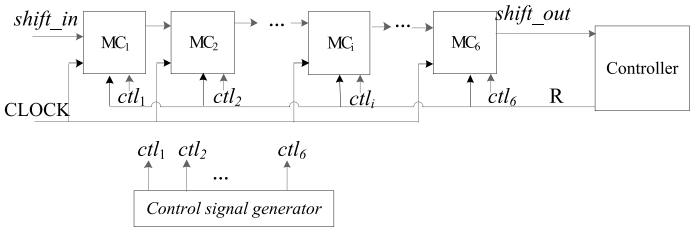


Fig. 5 The illustration of daisy chain access and the control signals generator

Each cell of the daisy chain is with a self-repair module, where two registers are contained, the original functional register DFF_1 and the redundant register DFF_2, for repair purposes. It will abandon the original DFF and use the redundant DFF replaced when a fault is detected in the MC. DFF_1 and DFF_2 can both receive data from *shift_in*, *Qout* is the output of the MUX_2, which chooses data from DFF_1 and DFF_2 (a MUX in the figure denotes a multiplexer with two inputs). The counter in the figure controls the repair of each MC. If an MC fails, the value of *Qout* differs from the expected response. The comparison result; signal *R* will be set to logic-1. The counter starts working when the falling edge of the clock comes. Then the value of the counter; *counterout* controls MUX_2 choosing the data bit of DFF_2.

If neither DFF_1 nor DFF_2 does work correctly, the counter will make a count operation on the failing edge of the clock. The overflow signal of the counter *ccarry* will be set to logic-1, which makes MUX_3 select data from *shift_in*. The failed MC is bypassed. The enable signal *counteren* is set to logic-0 simultaneously, which stops the counter work from then on. ctl_i is the access control signal of the MC_i, when ctl_i is set to logic-1, the CLOCK is enabled, and MC_i is accessed. Otherwise, MC_i is on hold. The new MC design proved efficient for daisy chain test and repair using test vectors before MEDA chips were used [33, 34]. Here we make a detailed analysis of the new design and propose a new test and repair scheme using the functional actuation patterns to ensure the bioassay's correctness in its running time.

Stuck-at faults cover a large percentage of faults and some unmodeled physical defects in COMS circuits [19], so we mainly consider the stuck-at faults here. In reality, all stuck-at faults of the daisy chain that appear in the path between MUX_1 and MUX_3 can be captured by the *Qout* signal. The wrong test response can activate the repair process of the MC. We use the stuck-at faults appearing in the DFF to represent the faults in the path between MUX_1 and MUX_3 across the DFF. In addition, the proposed design can use different numbers of redundant DFFs as long as the overall design area is less than the area of a microelectrode in a MEDA biochip.

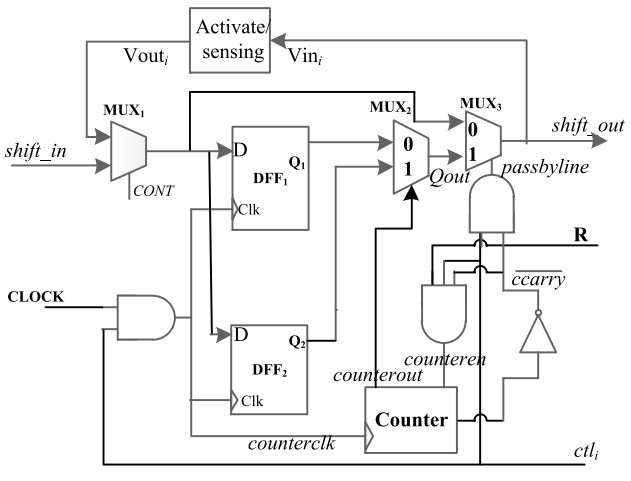


Fig. 6 The new MC with self-repair function

3.2 The Work Features of the MC with Self-Repair Module

As presented in Fig. 6, there are total six external signals; *CLOCK*, ctl_i, *R*, *shift_in* and *shift_out*, controlling the work of the new MC. For a new MC, a stuck-at fault may appear in the paths of DFF_1 or DFF_2, and DFF_1 and DFF_2 may be in S_0, S_1 or faultless states. Here S_0, S_1 and faultless denote stuck-at-0 fault, stuck-at-1 fault and with no fault. To summarize the work features of a new MC, the detailed work process of the new MC with different work states (Identified as T_1 to T_9) is shown in Table 1. Here, we use the State Transition Diagrams (STD) to present the running process of an MC, where the input/output denotes the shifted data bit (*shift_in*) and the production of the MC (*shift_out* or *Qout*).

The work STD of an MC in T_1 and T_4 is shown in Fig. 7. For T_1, If one data bit of logic-0 is shifted in, the fault is activated, and *R* is set as 1, which makes *shift_out* select the output of DFF_2 replace DFF_1. The MC is repaired successfully, and *Qout* outputs the target logic-0 (see transition (1)

Table 1 All work states situations of a MC

state	DFF_1/DFF_2	state	DFF_1/DFF_2
T_1	S_0/faultless	T_5	S_0/S_1
T_2	S_0/S_0	T_6	S_1/S_1
T_3	S_1/S_0	T_7	faultless/S_0
T_4	S_1/ faultless	T_8	faultless/S_1
T_9	faultless/faultless		

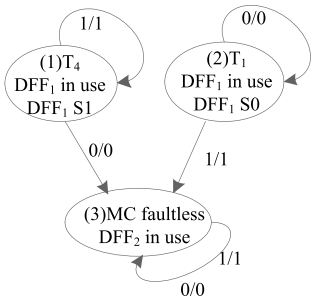


Fig. 7 Work STD for T1 and T4

to (2) in Fig. 7). Otherwise, if the data bit shifted in is logic-0 in state T_1, S_0 fault in DFF_1 cannot be activated. *Qout* still outputs the wrong path of DFF_1. However, the value is the target logic-0 for its stuck-at 0. The process is shown in the transition of (1) to (1). T_4 works in the same way as T_1. As shown in the transition (2) to (3), DFF_1 fault is activated, and the MC is repaired successfully. *Qout* outputs the target logic-0. In transition (2) to (2), the fault is not activated, *Qout* outputs the wrong path of DFF_1. It is also the right target logic-1.

The work STD of T_2 and T_3 is presented in Fig. 8(a). For T_2, first, the used register is DFF_1. If the data bit shifted in is logic-1, then S_1 fault in DFF_1 is activated. DFF_1 is replaced by DFF_2, *Qout* connects with DFF_2. Because DFF_2 has a S_0 fault, the output of *Qout* is logic-0. The process is described in the transition of (1) to (3). Otherwise, if logic-1 is shifted in T_2 state, S_1 in DFF_1 cannot be activated, *Qout* still connects to DFF_1. The value is the target logic-1. The process is shown by transition(1) to (1).

If MC is in (3), and the input data bit is logic-0, the fault cannot be activated, the output of the MC is the target logic-1. The process is shown in the transition (3) to (3). If the input bit is logic-1, S_0 in DFF_2 will be activated. The MC cannot be used afterwards, and it is bypassed, which is shown in the transition of (3) to (4). T_3 works in the similar way with T_2, and the work process is illustrated in states of (2), (3) and (4) in Fig. 8(a).

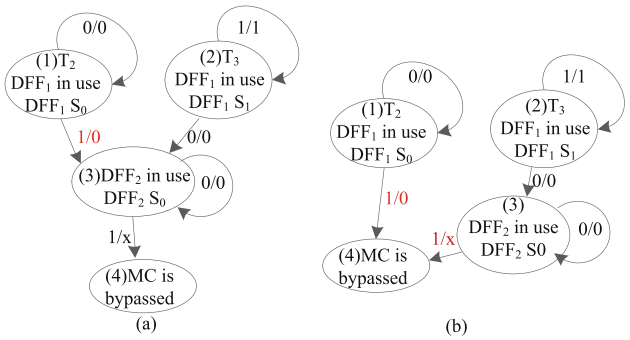


Fig. 8 Work STD T2 and T3. **a** work STD before MC soft-bypassed, **b** work STD after MC soft-bypassed

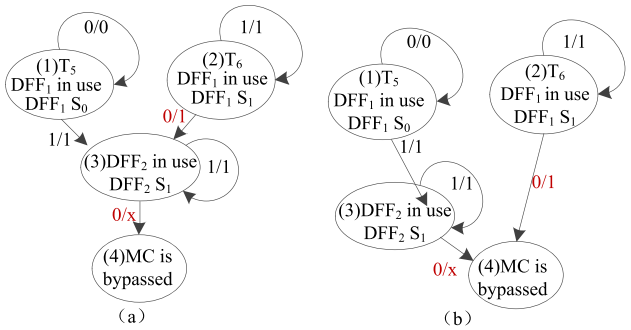


Fig. 9 Fig 10 Work STD T5 and T6. **a** work STD before MC soft-bypassed, **b** work STD after MC soft-bypassed

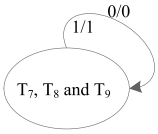
The work processes of T_5 and T_6 are given in Fig. 9. For T_5, DFF_1 is used first, if shifted bit is logic-0, the fault cannot be activate, the output is still logic-0 (see transition (1) to (1)). Otherwise, if input bit is 1, the S_0 fault in DFF_1 will be activated, *Qout* connect DFF_2 with S_1 fault instead of DFF_1, so the output of the MC is logic-1 (see transition (1) to (2)). After transition (1) to (2), DFF_2 is used in the MC. If input bit is logic-1, S_1 fault in DFF_2 will not be activated, the output of MC is still logic-1. The process is described by the transition (3) to (4). If input bit is logic-0, S_1 fault in DFF_2 will be activated, the MC cannot be used afterwards, and it was bypassed (see transition (3)to (4)). T_6 works in the same way with T_5, and the detailed STD is given in state(2), state(3) and state(4) in Fig. 9.

For fault T_7, T_8 and T_9, because DFF_1 is the Preferentially used register. Therefore, though fault in DFF_2 cannot be repaired when DFF_1 is faultless. However, the S_0 and S_1 fault in DFF_2 will have no impact on the daisy chain. The detailed STD is given in Fig. 10.

As presented above, in most cases, when data bit shifted in cannot activate the stuck fault, the fault value is same as the right target value at the output. When data bit shifted in can activate the fault, the MC is repaired, and the right target value also appears at the output. Therefore, for a shifted data bit, the value of output of a MC is same as the value of the input (such as 0/0 or 1/1 in the figures). This is same as a faultless MC in T_9 state given in Fig. 10. Thus, even though a fault with this kind appears, it has no effect on the running bioassay.

However, the input/output in the transitions (1) to (3) in Fig. 8 and (2) to (3) in Fig. 9 are different, the former is 1/0, and the latter is 0/1. The main function of registers in a MC is to shift one bit functional data to actuate or hold the corresponding droplet. The transitions (1) to (3) in Fig. 8 and (2) to (3) in Fig. 9 will change the shifted in data when shifted out, the changed

Fig. 10 Work STD of T7, T8 and T9



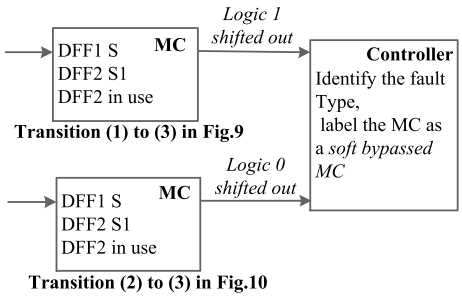


Fig. 11 Disposition of transitions (1) to (3) in Fig. 8 and (2) to (3) in Fig. 9

output would result in the wrong results of the running bioassay. Though these two kinds of faults hardly happen in real MEDA biochips, we also need to dispose it just in case. Here we use a software dispose solution; If a MC is in the transition of (1) to (3) in Fig. 8 or (2) to (3) in Fig. 9, we use the control soft program running in the controller to identify it, and label it as a soft bypassed MC. The illustration of the solution is show in Fig. 11, then the work STD of Figs. 8 and 9 can be described in Figs. 8(b) and 9(b).

Unlike a physically bypassed MC, a soft bypassed MC is just denoted as a MC that cannot be used again. It still can be accessed, and will be bypassed physically when the next data bit that can activate the fault shifted in.

3.3 Daisy Chain Self-Repair Using Functional Data in Running Process

As presented in Section 3.1, during each MC being accessed cycle, the data bit will also be sent to the controller to check the correctness. So we can use the controller to mark the work state of each MC. The detailed repair process is presented in this section.

3.3.1 Full Repair Coverage Using Actuation Patterns

The proposed scheme uses actuation patterns to test and repair the daisy chain. The flowchart of the detailed repair process of an MC is given in Fig. 12. As shown in the figure, an MC would be in one state of T_1-T_9 (given in Table 1) at a specific time. When a bit of actuation data is stored, the output of the MC will be compared with the correct response stored in the controller and gives feedback to control the repair process.

A data bit of an actuation pattern may be logic-0 or logic-1, the rhombus 1,6,15 and 22 all judge the value of the applied data bit as logic-0 or logic-1. They can activate different stuck-at faults. Other rhombuses in the figure all explain the compared process of *Qout* and the response stored in the controller. All other boxes present the different states of an MC.

All the daisy chain's MC states were unknown before actuation patterns shifted in. From the beginning, if the data bit shifted in is a logic-0 for the current MC, the output of the MC, *Qout*, is compared with the response stored in the controller. If they do not match, a stuck-at-1 fault is in DFF_1. *Qout* will receive data from DFF_2 replace DFF_1. The process is shown by rhombus 1 → rhombus 2 → box4. If the data bit shifted in current MC is logic-1, *Qout* and the response does not match, then DFF_1 is with an S_0 fault, and *Qout* connects to DFF_2, which is shown in rhombus 1 → rhombus 3 → box 5.

Either a data bit logic-0 or logic-1 is stored, and the output of MC will be compared with the stored response. If they match, we thought that the MC is faultless or with a stuck-at 1 (for data bit logic-1)/stuck-at 0 (for data bit logic 0). Then a new data bit is needed to identify the fault type. The data bit of logic-0 applied process is given by rhombus 1 → rhombus 2 → rhombus 6, and the data bit of logic-1 applied process is given by rhombus 1 → rhombus 3 → rhombus 6.

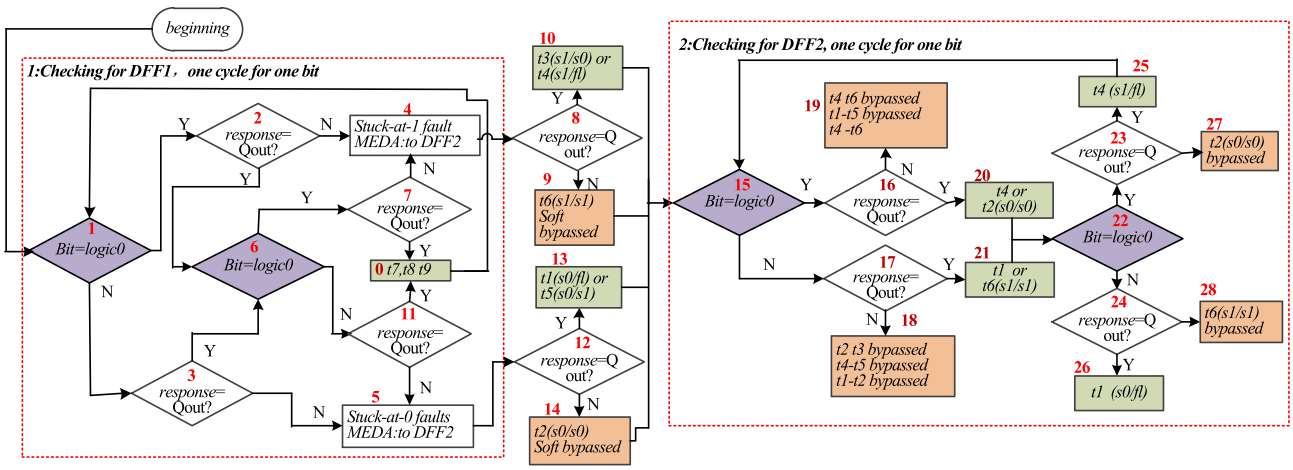


Fig. 12 Flowchart of the repair process of an MC

For rhombus 11, the data bit is logic-1, and *Qout* matches the response. The MC is recognized as a faultless MC. It may be in T_7, T_8 or T_9 state. The process is given by rhombus 11 → box 0. Otherwise, *Qout* and response do not match, and the MC is with a S_0 fault in DFF_1, and *Qout* shifts to DFF_2. The process is presented by rhombus 11 → box 5. Rhombus 7 works similarly to box rhombus 11, and it will go to box0 when *Qout* matches with the response and go to box 4 with stuck-at-1 existing in DFF_1.

When DFF_1 is replaced by DFF_2, *Qout*, which connects to DFF_2, is compared with the response in the same cycle. The process is explained by box 4 → rhombus 8 and box 5 → rhombus 12. In box 4 → rhombus 8, *Qout*, received data from DFF_2 is compared to the response. If they match, the MC can be recognized working in T_4 (S_1/S_0) or T_3 (S_0/S_1), which is presented in box 10. Another data bit needs to be applied to distinguish the state type. Otherwise, if *Qout* and response do not match, the MC will go to box 9. The MC is in T_6 state, where the MC is a soft bypassed MC. It will be bypassed physically when the next logic-0 is applied. Box 5 → rhombus 12 works in the same way. It may go to box 13, where MC is in T_1 or T_5 state or box 14 where MC is in T_2 state.

For box 10, box 13, box 9 and box 14, if a logic-0 is applied to the current MC, the response and *Qout* match, and the MC goes to box 20. Otherwise, if the response and *Qout* do not match, which said DFF_2 is with a stuck-1 fault, then T_5 in Box 13 and the soft bypassed MC in T_6 in box 9 will be bypassed physically, T_1 state in box 13 will become T_5, T_4 state in box 10 will become T_6, and they are all bypassed in box 19.

If the data bit shifted in logic-1, *Qout* and the response match, DFF_2 may be faultless (MC in T_1) or stuck at 1 (MC in T_6), shown in box 21. Otherwise, if they don't match, DFF is with a stuck-at-0 fault, and the state presented in box 10 will become T_3 and is bypassed physically. The soft bypassed MC in box 12 will be physically bypassed, too. The state in Box 10 will become T_5 state, the state in box 13 will transit to T_2 state in box 18, and both are physically bypassed.

The states presented in box 20 and box 21 need a new data bit to repair the fault or bypass the MC. If the data bit shifted in is a logic-0, then *Qout* is compared with the response in the same way. If they match, DFF_2 is faultless, and MC is in T_4 state shown by rhombus 22 → box 25. If they do not match, MC is in T_2 state, and it was bypassed. The process is demonstrated by rhombus 22 → box 27. If the data bit shifted in is logic-1, the MC works similarly; the MC will be recognized as an MC in T_1 or T_6 state and bypassed, which is shown by rhombus 22 → box 28.

A faultless MC may be in T_7, T_8, T_9 state, and it also may be in T_3 or T_1 state. See Fig. 12, no matter which states the MC is in, it will be recognized as a faultless MC or a bypassed MC when a new bit shifts in. If the MC is in T_7, T_8 and T_9 state, a logic-0 and a logic-1 can identify their work states and repair the detected fault. If the MC is in T_1, T_2, T_3, T_4, T_5 or T_6 state where DFF_1 is faulty, another pair of logic-0 and logic-1 must identify and bypass them.

Note that, 1) The pair of data bits of logic-0 and logic-1 can be logic-1 and logic-0, too. Therefore, for an MC, if there are two pairs of transitions between logic-0 and logic-1 applied, no matter what kind of faults exist in the MC, they can be repaired or bypassed. The two pair of transitions of logic-0 and logic-1 can be 0101, 0110, 1010 or 1001, called *the completed test data for a MC* to achieve 100% fault coverage. 2) Compared with other components of MC, DFFs are more prone to failure. Though we use stuck-at faults as an example of repaired faults, other faults that can make *Qout* different from the anticipated response can all be captured. However, only the faults that exist in the DFF branch path can be repaired.

3.3.2 Repair Process of the Daisy Chain Using Actuation Patterns

To describe the repair process more precisely, we present the work process of the daisy chain through four work modes. They are *actuating data shifting-in mode*, *actuation mode*, *sensing mode* and *sensed data shifting-out mode*.

In *actuation data shifting-in mode*, actuation patterns are shifted in the daisy chain serially. When 1-bit actuation data shifts in the register for each MC, it will be compared with the anticipated response. The new MC is fault-free if they match, and signal *R* is logic-0. Otherwise, the indicating signal *R* is set as 1, activating the self-repair process as presented in Section 3.2. If an MC is bypassed or soft bypassed, it will influence the running bioassay, and then an adaptive online synthesis algorithm can be developed to adjust the execution of operations in a real-time manner to avoid the bypassed MCs.

Figure 13 gives a daisy chain with 16 MCs. Each MC is in a state described in Table 1. See the Figure, MC_1, MC_5, MC_6, MC_7 and MC_{12} are in state T_9, T_7, T_9, T_9 and T_8, respectively. These five MCs are all faultless. For MC_2, MC_8, MC_9 and MC_{11}, they are in state T_1 where S_0 is in DFF_1. MC_{15} is in state T_2, whose DFF_1 and DFF_2 both stuck at 0. MC_{14} is in state T_3, whose DFF_1 stuck at logic 1 and DFF_2 stuck at 0. MC_3, MC_4 and MC_{10} are in state T_4, and their DFF_1s stuck at 1. MC_{13} is in state T_5; its DFF_1 stuck at 0 and DFF_2 stuck at 1. MC_{16} in state T_6 whose DFF_1 and DFF_2 stuck at 1 and 0, respectively.

Now, we assume that a 2 × 2 droplet is maintained by MC_1, MC_2, MC_5 and MC_6 and moves right. First, to maintain the droplet, the actuation pattern A1, 1100 1100 0000 0000, needs to be applied to the daisy chain, and four work modes will be undergone.

1. *Actuation data shifting-in mode*

   As shown in Fig. 14, MC_1, MC_5, MC_6, MC_7, MC_{12} are faultless, they work normally. For MC_2, the bit shift is logic-1, stuck-at 0 in DFF_1 of state T_1 is activated. The

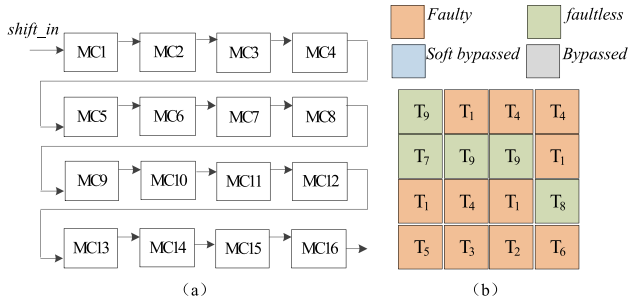


Fig. 13 A daisy chain with 16 MCs

shifted in bits are all logic-0 s for MC_3, MC_4 and MC_{10}. So stuck-at-1 in DFF_1 in state T_4 are activated and repaired successfully. The right bits are stored in them successfully. For T_1 in MC_9 and MC_{11}, though the shifted bits of logic-0 s cannot activate the corresponding faults, the right value can still be read from these MCs. MC_{13} and MC_{15} both can read the right data in the same way.

For T_6 in MC_{16}, the stuck-at-1 fault in DFF_1 is activated by the shift in logic-1. However, the output of MC_{16} is the wrong bit logic-0 because of the stuck-at-0 fault at DFF_2. It will be denoted as a soft bypassed MC and cannot be used from then on. For T_3 in MC_{14}, the current fault is activated, and the output of MC with T_3 is the right data bit. For T_2 in MC_{15} and T_5 in MC_{13}, the faults are all not activated but work appropriately now. After A1 is shifted, the correct data of A1 is stored in the daisy chain.

2. *Actuating mode*

   A1 will hold the droplet at the 2×2 location consisting of MC_1, MC_2, MC_5, and MC_6.

3. *Sensing mode*

   In the sensing mode, the droplet locations are sensed and stored in the register of each DFF. The sensed pattern is the same as the actuation pattern if no other electrode faults exist.

4. *Shifting out mode*

   To check the correctness of the bioassay, the data shifted out cannot be changed, so all sensed data must be shifted out directly. So, in this mode, all control signals of *ctl*s are set as logic 1, and sensed data are shifted out to the Controller serially.

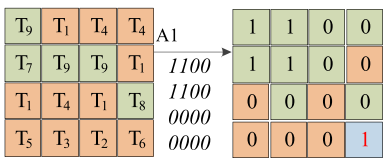


Fig. 14 The daisy chain is repaired in the actuation pattern 1100110000000000 shifted in

To move the 2×2 droplet right, the next actuation pattern A2 is shifted in the daisy chain. While A2 is applied, the daisy chain is repaired similarly, where MC_{16} cannot be used, and the corresponding data bit is denoted as don't care bit x. Then the droplet continues to move right, and A3 is applied. A2 and A3 repair results are shown in Fig. 15. We can see that, MC_2, MC_3, MC_4, MC_8, MC_{10}, MC_{11} are all repaired successfully, MC_{16} is bypassed at last. MC_9 is not repaired but has no impact on the running of the bioassay, and it will be repaired when data bit 1 is stored. Other MCs whose DFF_1 and DFF_2 are both faulty will be denoted as soft bypassed MCs and bypassed when the adaptive data bits shift in.

4 Simulational Results

This section performed extensive experiments to estimate the proposed online daisy repair scheme. A high-level synthesis algorithm [16] was used to generate actuation patterns. Three real-life benchmarks, CEP, serial dilution, and master mix [5] were used here. CEP combines three small bioassays: cell lysis, mRNA extraction, and mRNA purification. The biochip with a size of 30×60 was used. The droplet size is set as 4×4.

4.1 Actuation Patterns Generating and Analyzed

In order to repair more MCs in the operation of the MEDA biochip, some statistics are performed on the actuation patterns of the bioassay benchmarks. The high-level synthesis

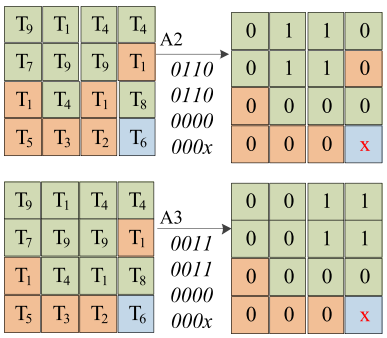


Fig. 15 The daisy chain is repaired in the actuation pattern A2 and A3 shifted in

Table 2 The actuation patterns statistics results

benchmarks	N	MCu	N_1	N_t	$ratio_1$
CEP	47	29	634	8.5k	0.7%
Master-mix	35	21	1412	6.3k	2.2%
Serial_dilution	79	52	2680	14.2k	1.9%

algorithm [16] is used to generate the corresponding actuation patterns. The actuation patterns statistic results are given in Table 2. Three benchmarks are listed in the first column. The second column N is the number of actuation patterns. MCu denotes the number of MCs that used by the bioassay. The last three columns N_1, N_t and $ratio_1$ are the number of 1 in the total actuation patterns, the number of whole data bits, and the ratio of N_1 and N_t computed as N_1/N_t, respectively. From the table, we can conclude that: 1) The actuation patterns are very sparse. For benchmark CEP, the percentage of the number of bit 1 only takes up 0.7% of the total actuation bits. 2)A number of MCs have not been used in the execution of the bioassays. The number of used MCs of CEP, Master-mix, and Serial-dilution are only 29, 21 and 52, respectively.

4.2 Using Actuation Patterns to Repair Daisy Chain During Bioassay's Running

By summing up the features of the actuation patterns, we found that only a tiny proportion of MCs are used in the execution of the bioassay. Other MCs that did not be used are labelled as unoccupied MCs. To ensure the fluidic operations' accuracy, all fluidic operations should abide by fluid constraints, where the distance between two fluidic operations should keep at least four unused MC spaces here. In the synthesis algorithm, we set all values of actuation patterns of the unoccupied MCs as 0. The actuation patterns set is denoted as Apu0. The states in Table 1 are classified into three types; fault free type *Frt* that contains T_1, T_7 and T_9, repairable type *Rt* including T_1 and T_4 and bypassed type *Bt* including T_2, T_3, T_5 and T_6. Some *Rt* and *Bt* faults are randomly inserted in the daisy chain of benchmarks. Then Apu0 is applied to the new daisy chain. The experimental results are given in Table 3.

The benchmarks are listed in the first column. As presented in Section 3.3.1, if the completed test vector is applied to an MC, no matter what faults can be repaired or bypassed. The number of MCs with the completed vector applied is denoted as *MCa*. *F_Rt* and *F_Bt* in the table denote the number fault types *Rt* and *Bt*, respectively. $in = 4$ denotes four randomly inserted faults in MCs with completed test vectors. *out* in the table denotes faults inserted in other kinds of MCs. The last three columns are *R_Rt* denoting the number of repaired *Rt* faults, *R_Bt* denoting the number of repaired *Bt* faults and FC denoting the fault coverage.

For faults inserted in MCs with completed test vectors, they indeed can be repaired or bypassed. However, the faults cannot be repaired or bypassed for the MCs with no completed vectors. The table shows that the inserted defect for benchmark cep, *F_Rt*, is $4 + 6 = 10$, fortunately, repaired successfully, where *R_Rt*=10. However, for the benchmark Master-mix, the inserted *F_Rt* is 5+7=12, and the five faults in the MC with completed test vectors can be repaired or bypassed. But seven faults inserted in other MCs repaired only 5, where *F_Rt*=10. For benchmark serial-dilution, *F_Rt* is $4 + 6 = 10$, and they are all repair successful, where $F\_Rt = 8$. *Bt* faults are also bypassed or soft bypassed in the same way. For benchmark CEP, inserted *Bt*; $F\_Bt = 0+1$, and the number of bypassed *Bt*; $R\_Bt = 0$. *Bt* of benchmark master-mix is 1+1, and the number of bypassed *Bt*, *R_Bt*, equals 1. Benchmark serial-dilution has the same result as benchmark CEP. The last column of the table is fault coverage computed as $(R\_Rt + F\_Bt)/(F\_Rt + F\_Bt)$. Because some MC was not identified, faults in all three benchmarks are not repaired or bypassed by the actuation patterns Ap0. Note they do not affect the running bioassay.

4.3 Fill Spare MC Using Suitable Data

As the results presented in Section 4.1, the actuation patterns may not cover all faults in the daisy chain. Here, we fully use the spare electrodes in the bioassay execution. Because the value of the spare electrodes does not affect the execution of the bioassay, they can be filled with suitable data bits to repair the corresponding MC.

Table 3 The actuation patterns statical results

Benchmarks	*MCa*	*F_Rt*		*F_Bt*		*R_Rt*	*R_Bt*	FC
		in	*out*	*in*	*out*			
cep	29	4	6	0	1	10	0	90.91%
master-mix	20	5	7	1	1	10	1	78.57%
serial-dilution	50	2	6	0	1	8	0	88.89%

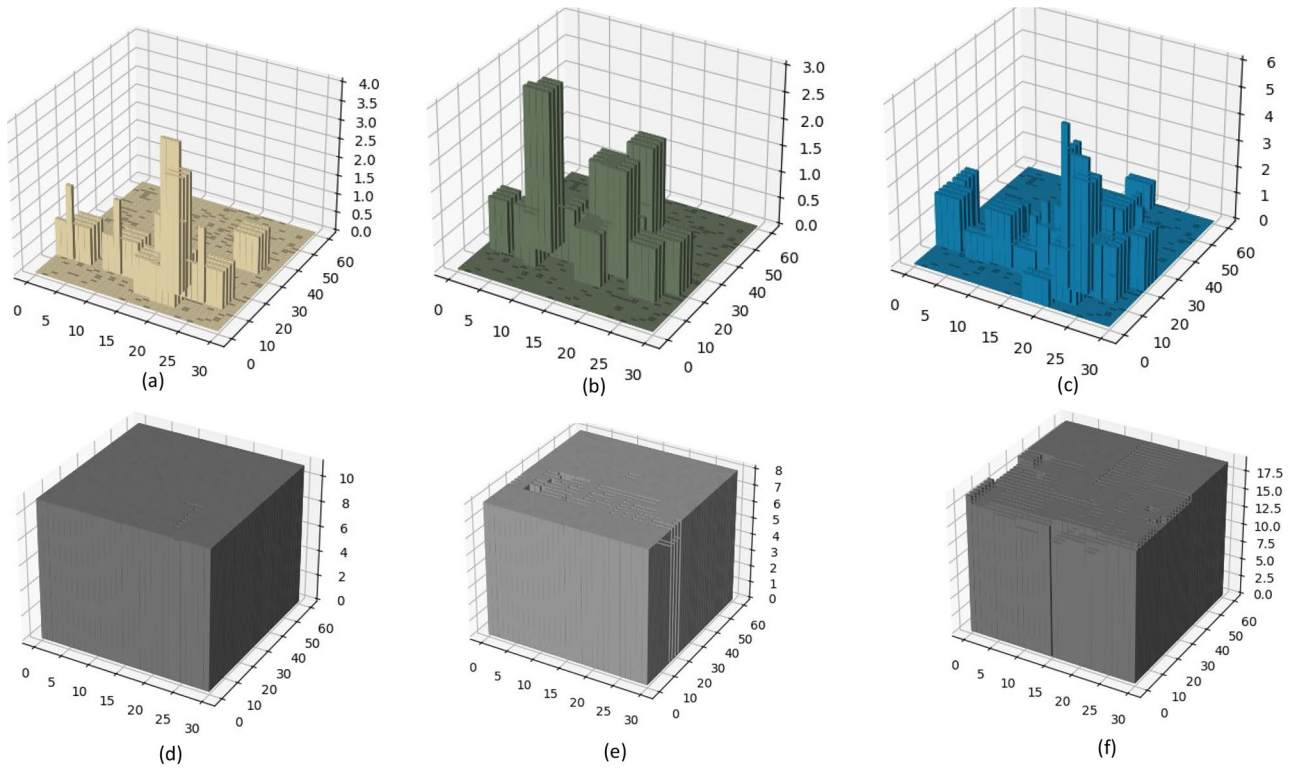


Fig. 16 The number of completed test vectors before and after don't care bits of spare electrodes filled

Here, filling the don't care bits of the spare MCs in actuation patterns abides by the rules of generating as many completed test vectors for each MC as possible. To explore the difference in the number of completed test vectors for each MC before and after filling the don't care bits. We make care filling process and count the corresponding number of completed test vectors for each MC. The results are presented in. Fig 16. See the figure; before the don't care bits are filled in the actuation patterns, the number of completed vectors is minimal (see (a), (b) and (c)). The number of completed vectors is a lot more after don't care bits filled with suitable data (see (e), (f) and (g)). It is said that if some faults occur in the daisy chain, they are more likely to be repaired or bypassed without effect on the running bioassay.

We also insert some faults in different MCs in different bioassay execution clock cycles. the experimental results are given in Table. 4. As shown in the table, the second and the third column are also the F_Rt and F_Bt, denoting the number of inserted faults. The fourth and the fifth column are the numbers of repaired faults; *R_Rt* and *R_Bt*. The last column is the fault coverage. We can see that because many spare electrodes can be used, all MCs can get the completed test vectors, and all inserted faults can be repaired or bypassed. The fault coverage is all 100%.

For big-scale bioassays, the electronics would be used more, and unoccupied electrodes will be very few even has no unoccupied electrodes. Then the MCs may be repaired completely by the primary actuation patterns. So, we can use adapt solutions for different scale bioassays in reality.

4.4 Discussion

As presented in Section 3.3.1, if two pairs of transitions between logic-0 and logic-1 are applied to an MC, no matter what stuck-at faults exist, they can be repaired or bypassed. The two pair of transitions of logic-0 and logic-1 can be 0101, 0110, 1010 or 1001, which are called the completed test data for an MC to achieve 100% fault coverage.

The actuation patterns may not cover all faults in the daisy chain in the bioassay execution. The spare electrodes does not affect the execution of the bioassay, they can be filled with suitable data bits to generate more completed test data to repair the corresponding MC. The simulation results

Table 4 The experimental results with spare electrodes filled with suitable data

Benchmarks	*F_Rt*	*F_Bt*	*R_Rt*	*R_Bt*	FC
cep	10	1	10	1	100%
master-mix	15	2	15	2	100%
serial-dilution	15	1	15	1	100%

prove the number of completed test vector is very more after fill adapt data in spare electrodes. Therefore, the actuation patterns can repair the daisy chain with a 100% fault coverage in most cases in the bioassay running.

The limitation of the method is that when an MC cannot be repaired, it will be bypassed and influence the running bioassay; then, an adaptive online synthesis algorithm must be developed to adjust the execution of operations in a real-time manner to avoid the bypassed MCs.

This method tests, diagnoses and repairs the daisy chain using bioassay's actuation patterns. During the bioassay execution, the corresponding actuation pattern must be shifted into the 1-bit registers of MCs through the daisy chain. The proposed scheme performs the test, diagnose and repair function in the shifting process. Therefore, the proposed online daisy chain repair scheme didn't take up any additional time. Furthermore, this method does not produce additional hardware costs compared to the literature [34] and does not contribute to electrode degradation.

5 Conclusion

Daisy chain is the key part of a MEDA biochip, the reliability is critical. The paper has proposed a new daisy chain self repair method for MEDA biochips. The daisy chain can be repaired in the actuation patterns shifted in through the new serial visiting scheme and fault tolerant module. The proposed scheme can test each MC, and capture all faults causing wrong response in real time. If wrong actuation data bit is caused by DFF1 branch path or DFF2 branch path, the MC can be repaired or bypassed in the following accessed time. Test data used is actuation patterns, test and repair were completed during actuation patterns shifted in, which did not take any additional time and without causing the electrode's degradation. Test and repair process had no any effect on the executing bioassay. The experimental results also proved the effectiveness.

Funding The research work was supported by the National Natural Science Foundation of China (No.62241404) and the Scientific Research Foundation of the Educational Commission of Jiangxi Province (No. GJJ210501).

Data Availability This manuscript has no associated data.

Declarations

Conflict of Interest The author had received funding for this research from the following sources National Natural Science Foundation of China (No.62241404) and the Scientific Research Foundation of the Educational Commission of Jiangxi Province (No. GJJ210501). The author has no relevant financial or non-financial interests to disclose. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Babies (2021) Babies. https://baebies.com/
2. Chakrabarty K (2009) Design automation and test solutions for digital microfluidic biochips. IEEE Trans Circ Syst I Reg Papers 57(1):4–17
3. Choi K, Ng AH, Fobel R, Wheeler AR (2012) Digital microfluidics. Ann Rev Anal Chem 5:413–440
4. Dong C, Liu L, Liu H, Guo W, Huang X, Lian S, Liu X, Ho T-Y (2020) A survey of dmfbs security: State-of-the-art attack and defense. In Proceedings of the 2020 21st International Symposium on Quality Electronic Design (ISQED), pp 14–20. IEEE
5. Elfar M, Zhong Z, Li Z, Chakrabarty K, Pajic M (2017) Synthesis of error-recovery protocols for micro-electrode-dot-array digital microfluidic biochips. ACM Trans Embed Comput Syst (TECS) 16(5s):1–22
6. Guo W, Lian S, Dong C, Chen Z, Huang X (2022) A survey on security of digital microfluidic biochips: Technology, attack, and defense. ACM Trans Des Automat Electron Syst (TODAES) 27(4):1–33
7. Ho Y, Wang G, Lai KY-T, Lu Y-W, Liu K-M, Wang Y-M, Lee C-Y (2016) Design of a micro-electrode cell for programmable lab-on-cmos platform. In: Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), pp 2871–2874. IEEE
8. Huang H-C, Liang C-C, Wang Q, Huang X, Ho T-Y, Kim C-J (2022) Nr-router: Non-regular electrode routing with optimal pin selection for electrowetting-on-dielectric chips. In: Proceedings of the 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), pp 56–61. IEEE
9. Huang X, Liang C-C, Li J, Ho T-Y, Kim C-J (2019) Open-source incubation ecosystem for digital microfluidics-status and roadmap. In: Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 1–6. IEEE
10. Ibrahim M, Li Z, Chakrabarty K (2015) Advances in design automation techniques for digital-microfluidic biochips. In: Proceedings of Formal Modeling and Verification of Cyber-Physical Systems, pp 190–223. Springer
11. Illumina (2021) Illumina. https://www.illumina.com.cn/science/technology/digital-microfluidics.html
12. Lai KY-T, Yang Y-T, Lee C-Y (2015) An intelligent digital microfluidic processor for biomedical detection. J Signal Process Syst 78(1):85–93
13. Li Zipeng, Lai Kelvin Yi-Tse, Yu Po-Hsien, Chakrabarty Krishnendu, Ho Tsung-Yi, Lee Chen-Yi (2017) Structural and functional test methods for micro-electrode-dot-array digital microfluidic biochips. IEEE Trans Comput Aided Des Integr Circ Syst 37(5):968–981
14. Li Z, Lai KY-T, McCrone J, Yu P-H, Chakrabarty K, Pajic M, Ho T-Y, Lee C-Y (2017) Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip. IEEE Trans Comput Aided Des Integr Circ Syst 37(3):601–614
15. Liang T-C, Zhou J, Chan Y-S, Ho T-Y, Chakrabarty K, Lee C (2021) Parallel droplet control in meda biochips using multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning, pp 6588–6599. PMLR
16. Li Z, Ho T-Y, Lai KY-T, Chakrabarty K, Yu P-H, Lee C-Y (2016) High-level synthesis for micro-electrode-dot-array digital microfluidic biochips. In: Proceedings of the 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp 1–6. IEEE
17. Li Z, Lai KY-T, Yu P-H, Chakrabarty K, Ho T-Y, Lee C-Y (2016) Built-in self-test for micro-electrode-dot-array digital microfluidic biochips. In: Proceedings of the 2016 IEEE International Test Conference (ITC), pp 1–10. IEEE

18. Miroculus (2021) Miroculus. https://miroculus.com/company/
19. Reddy C, SaiTulasi K, Anuja T, Rajarajeswari R, Mohan N (2021) Automatic test pattern generation of multiple stuck-at faults using test patterns of single stuck-at faults. In: Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pp 71–75. IEEE
20. Ryu K, Jung J, Jung D-H, Kim JH, Jung S-O (2015) High-speed, low-power, and highly reliable frequency multiplier for dll-based clock generator. IEEE Trans Very Large Scale Integr (VLSI) Syst 24(4):1484–1492
21. Samiei E, Tabrizian M, Hoorfar M (2016) A review of digital microfluidics as portable platforms for lab-on a-chip applications. Lab Chip 16(13):2376–2396
22. Shukla V, Hussin FA, Hamid NH, Zain Ali NB (2017) Advances in testing techniques for digital microfluidic biochips. Sensors 17(8):1719
23. Shukla V, Hussin FA, Hamid NH, Ali NBZ, Chakrabarty K (2017) Offline error detection in meda-based digital microfluidic biochips using oscillation-based testing methodology. J Electron Test 33(5):621–635
24. Shukla V, Ali NBBZ, Hussin FA, Hamid NHB (2014) Diagonal testing in digital microfluidics biochips using meda based approach. In: Proceedings of the 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS), pp 1–5. IEEE
25. Shukla V, Ali NBBZ, Hussin FA, Hamid NH, Sheikh MA (2016) Fault modeling and simulation of meda based digital microfluidics biochips. In: Proceedings of the 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), pp 469–474. IEEE
26. Shukla V, Ali NBZ, Hussin FA, Zwolinski M (2013) On testing of meda based digital microfluidics biochips. In: Proceedings of the Fifth Asia Symposium on Quality Electronic Design (ASQED 2013), pp 60–65. IEEE
27. Shukla V, Hussin FA, Hamid NH, Ali NBBZ (2016) Investigation of capacitance dependence on droplet volume in meda based biochips. In: Proceedings of the 2016 6th International Conference on Intelligent and Advanced Systems (ICIAS), pp 1–5. IEEE
28. Wang G (2013) Field-programmable microfluidic test platform for point-of-care diagnostics. PhD thesis, University of Saskatchewan
29. Wang G, Teng D, Lai Y-T, Lu Y-W, Ho Y, Lee C-Y (2014) Field-programmable lab-on-a-chip based on microelectrode dot array architecture. IET Nanobiotechnol 8(3):163–171
30. Xu T, Chakrabarty K (2009) Fault modeling and functional test methods for digital microfluidic biochips. IEEE Trans Biomed Circ Syst 3(4):241–253
31. Xu T, Chakrabarty K (2007) Functional testing of digital microfluidic biochips. In: Proceedings of the 2007 IEEE International Test Conference, pp 1–10. IEEE
32. Yang J-S, Huang S-Y (2005) Quick scan chain diagnosis using signal profiling. In: Proceedings of the 2005 International Conference on Computer Design, pp 157–160. IEEE
33. Zhang L, Li Z, Chakrabarty K (2018) Built-in self-diagnosis and fault-tolerant daisy-chain design in meda biochips. In: Proceedings of the 2018 IEEE International Test Conference (ITC), pp 1–10. IEEE
34. Zhang L, Li Z, Huang X, Chakrabarty K (2023) Enhanced built-in self-diagnosis and self-repair techniques for daisy-chain design in meda digital microfluidic biochips. IEEE Trans Comput Aided Des Integr Circ Syst
35. Zhong Z, Chakrabarty K (2020) Ijtag-based fault recovery and robust microelectrode-cell design for meda biochips. IEEE Trans Comput Aided Des Integr Circ Syst 39(12):4921–4934
36. Zhong Z, Li Z, Chakrabarty K, Ho T-Y, Lee C-Y (2018) Microelectrode-dot-array digital microfluidic biochips: Technology, design automation, and test techniques. IEEE Trans Biomed Circ Syst 13(2):292–313
37. Zhong Z, Li Z, Chakrabarty K (2018) Adaptive and roll-forward error recovery in meda biochips based on droplet-aliquot operations and predictive analysis. IEEE Trans Multi-Scale Comput Syst 4(4):577–592
38. Zhong Z, Chakrabarty K (2019) Fault recovery in micro-electrode-dot-array digital microfluidic biochips using an ijtag network behaviors. In: Proceedings of the 2019 IEEE International Test Conference (ITC), pp 1–10. IEEE
39. Zhong Z, Liang T-C, Chakrabarty K (2020) Enhancing the reliability of meda biochips using ijtag and wear leveling. IEEE Trans Comput Aided Des Integr Circ Syst
40. Zhong Z, Li Z, Chakrabarty K (2017) Adaptive error recovery in meda biochips based on droplet-aliquot operations and predictive analysis. In: Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 615–622. IEEE

Ling Zhang received the bachelors degree in computer technology and Science from the Huaibei Normal University, Huaibei, China, in 2003 and the masters and Ph.D. degree in Computer Science from Hunan University, Changsha, China, in 2006, and 2013. She is an associate professor with the Department of Software and Internet of Things Engineering of Jiangxi University of Finance and Economics, Nanchang, China. Her research interests include testing and fault tolerance for microfluidic biochips and integrated circuits.