Design Automation for Micro-Electrode-Dot-Array Based Digital Microfluidic Biochips

A DISSERTATION

Submitted in partial fulfillment of the

requirements for the award of degree

of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

SURBHI PAL

(Enrollment No.: 17535028)

Under the guidance of

Dr. Sudip Roy

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE - 247667 (INDIA)

MAY, 2019

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation "**Design Automation for Micro-Electrode-Dot-Array Based Digital Microfluidic Biochips**" towards the fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science and Engineering**, submitted to the Department of Computer Science and Engineering, **Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during May, 2018 to May, 2019 under the guidance of **Dr. Sudip Roy**, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee.
The content presented in this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date:
Place: Roorkee

(Surbhi Pal)

CERTIFICATE

This is to certify that the statement made by the candidate in declaration is correct to the best of my knowledge and belief.

Date:
Place: Roorkee

(**Dr. Sudip Roy**)
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee

ABSTRACT

Microfluidics is the branch of science which deals with the manipulation of fluids in the sub-millimeter scale. With the evolution of technology, digital microfluidic (DMF) biochips have become a vital part of biochemical research. It is a very small chip that performs thousands of operations at the same time. It has a great impact on biomedical and biochemical science. With the evolution in the fabrication technology, efficient microfluidic biochip technologies have been proposed. The chip deals with performing operations on very small volume fluids (like microlitre, nanolitre, picolitre, femtolitre). It performs droplet based operations like sample preparation, mixing, routing, etc. Micro-electrode-dot-array digital microfluidic (MEDA-DMF) biochips have advance features over traditional DMF biochips. In this dissertation, we talk about one-pass synthesis approach for MEDA-DMF biochip. We propose a routing algorithm for MEDA-DMF biochip which is based on the concepts of computational geometry. We also propose well-defined library, architecture and bioassay files which can be used as input for synthesis of biochip. We also decided to develop a tool for MEDA-DMF biochip.

ACKNOWLEDGMENTS

I would never have been able to complete my dissertation without the guidance of my supervisor, and support from my family and friends.

I would like to express a thankful note to my supervisor, **Dr. Sudip Roy**, for his guidance. I am also grateful to the Department of Computer Science and Engineering of Indian Institute of Technology Roorkee for providing valuable resources to aid my research. I would like to thank PhD student **Mr. Debraj Kundu** and all the lab colleagues who were always willing to help and give their best suggestions.

A hearty thanks to **my parents** for encouraging me in good and bad times.

Surbhi Pal

CONTENTS

List of Figures

Chapter 1

INTRODUCTION

Till date many devices have been developed to simplify the biochemical laboratory tasks and one of the such devices is biochip (also known as lab-on-a-chip). In traditional laboratories, the bioassay protocol steps are done manually using test tubes for mixing and microscope or other detectors for analyzing the output. In order to make this whole process automated requiring no human intervention and to reduce the space requirement and cost, biochips are used. Biochips make the process faster as compared to traditional test tube-based laboratories. The research domain of biochip is an interdisciplinary area including biotechnology, mechanical engineering, chemical engineering, electronics and computer science. There are different kinds of biochips developed till now, some of them are given below:

- Implantable chip: This chip can be implanted in the body of any animal or human being. It can be used to trace the person, uniquely identify him/her and rescue a sick person.
- Implantable biosensor: It can be used to monitor glucose in the human body.
- Microarray: It is a two dimensional grid which has a higher density of features per unit area. It is neither reconfigurable nor scalable. It is of two types: Protein array and DNA array.
  - DNA array: This microarray contains DNA pieces affixed to a solid surface. It affixed DNA segments are called probes.
  - Protein microarray: It is a miniature array having a huge number of different capture agents deposited on a chip surface to determine the presence and amount of proteins in biological samples.
- Microfluidic biochips: These biochips works with the small amount of fluid and its popularity is increasing day-by-day. There are two types of microfluidic biochips, namely, digital microfluidic biochip and continuous flow microfluidic biochip.
  - Digital microfluidic (DMF) biochip: It works with the principle of EWOD (electrowetting-on-dielectric) using small (micro, nano, or picolitre volume) droplets of biochemical fluids [1].

- Continuous flow microfluidic (CMF) biochip: Its operation is based on the continuous flow of fluid [1, 2].

In *DMF* biochip synthesis, the main aim is to produce actuation sequence from the bio description protocol requiring limited number of resources and minimum time. The input biochemical samples are given to the dispensing ports (input reservoirs) and the actuation sequence controls the movement of droplets. The detectors are attached to the chip where output droplets are transported. To find the actuation sequence many intermediate steps are followed, i.e., scheduling, resource binding, module placement and droplet routing. Time optimization is the responsibility of scheduling and droplet routing steps. Resource optimization is the aim of resource binding and module placement. All these steps are designed as software which is embedded on the microcontroller of the biochip.

DMF biochip has several limitations, like lack of integrated sensors, constraints on droplet volume and size, etc. To resolve the limitations of DMF Boichip, MEDA-DMF biochip architecture was proposed. MEDA-DMF biochip architecture is based on the sea-of-micro-electrodes concept. Each microelectrode cell consists of a microelectrode, a sensing circuit and an activation circuit. In this architecture, diagonal movement of droplets is allowed, integrated sensors are present and size, shape and volume of droplet are controllable.

1.1 Contribution of Dissertation

In this dissertation, a new Synthesis approach has been proposed for MEDA-DMF biochip. The objective of the design automation or the synthesis of a microfluidic biochip is to efficiently map a sequencing graph of a given protocol onto the biochip. Extensive research work has been published on the design automation for DMF biochips but due to the architectural differences design automation for MEDA-DMF biochip requires a completely different approach. A one-pass synthesis approach is presented here for MEDA-DMF biochip where the placement of the microfluidic modules and the corresponding routing of the droplets are calculated simultaneously. Initially a best-fit microfluidic module mapping is performed with a prioritized operation node of the sequencing graph. Next, all possible free-rectangles are calculated from the current configuration of the MEDA-DMF biochip. Among those free rectangles, the best-fit free-rectangle is chosen for the module placement and simultaneously the routing paths for it are also calculated.

1.2 Organization of Dissertation

This dissertation has been divided in chapters.

Chapter 1 provides introduction of microfluidic biochip, its applications and contribution of this dissertation.

Chapter 2 describes the layout of DMF biochip, CMF biochip and MEDA-DMF biochip. The introduction about reconfigurable and non-reconfigurable resources is also given here.

Chapter 3 provides the brief introduction of different scheduling, placement and routing algorithms. This chapter also explains the flow of synthesis.

Chapter 4 lists research gaps and discusses exact problem statement.

Chapter 5 explains the types of operations in MEDA-DMF biochip and input files.

Chapter 6 explains the proposed scheduling, placement and routing approach.

Chapter 7 shows the empirical results generated.

Chapter 8 concludes the dissertation and provides future scope in this work.

Chapter 2

BASIC PRELIMINARIES

In the following sections, the architectures of CMF, DMF and MEDA-DMF biochips have been explained.

2.1 Continuous Flow Microfluidic Biochips

Continuous flow microfluidic biochip is the first generation of microfluidic biochips [1, 2]. Its operation is based on continuous flow of fluid. CMF biochip contains micropumps, microvalves and microchannels. Flow of liquid can be triggered by using external pressure, electro kinetic mechanism or integrated mechanical micro-pumps. Electroosmosis (a common electro kinetic method) is the motion of ionic fluid solution using electrical field. CMF biochips are not suitable for complex tasks that require complicated fluid manipulations. They are difficult to integrate and scale.

Figure 2.1 shows the architectural layout of CMF biochip with number of mixers, $N_m = 1$, number of input reservoirs, $R_n = 4$ and number of storage units, $q = 2$. In CMF biochip, the k-segment rotary mixer is used for the mixing operations.

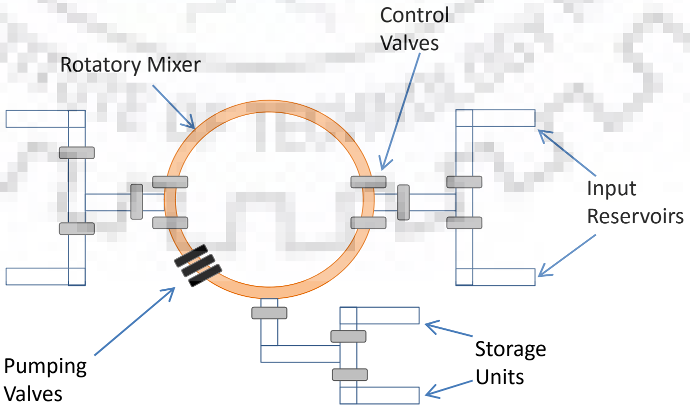


Figure 2.1: Architectural layout of CMF biochip.

2.2 Digital Microfluidic Biochips

DMF biochip is the second generation of microfluidic biochips [1]. These biochips manipulate the liquid as discrete droplets. DMF biochips offer more scalable system architecture compared to CMF biochips. They have the capability of dynamic reconfiguration.

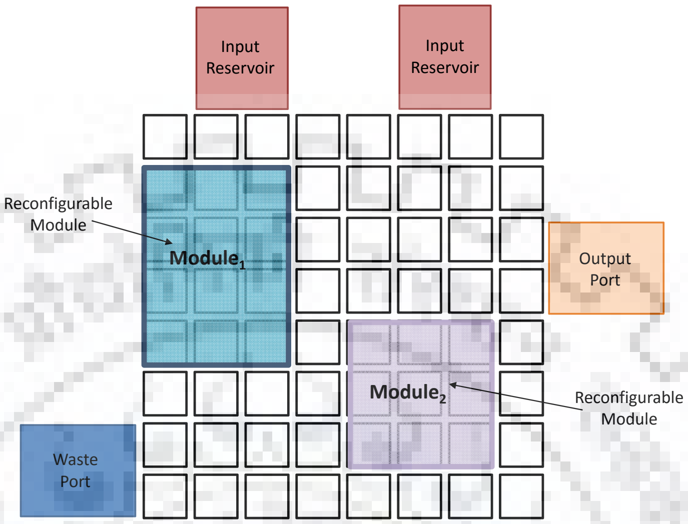


Figure 2.2: Architectural layout of DMF biochip.

A DMF biochip is a 2-D cell array of electrodes as shown in Figure 2.2. Each cell consists of two parallel glass plates. Bottom plate contains control electrodes and top plate contains ground electrode. A dielectric insulator is added to the top and bottom plate for adding the capacitance between electrode and droplet. Droplet containing biochemical sample is sandwiched between the two plates. DMF biochip consists of microfluidic array, dispensing ports, output detectors to analyze the output and on-chip mixers as shown in Figure 2.2. In order to move a droplet, the electrode adjacent to the droplet is activated by applying control voltage and electrode under the droplet is turned off. Droplet will move towards the activated electrode due to EWOD (Electrowetting-on-dielectric) effect. EWOD causes charge to accumulate in droplet-insulator interface which in turn creates inter-facial tension gradient across the gap between these two adjacent electrodes. This process causes the droplet to move. Actuation sequence defines which electrode will be activated. Suppose droplet has to move from electrode E1 to E2. Deactivate E1, i.e., reset it to 0 and activate E2, i.e., set it to 1. At each time interval, all the electrodes are coded in the form of 0 and 1.

2.3 Micro-electrode-dot-array Based Digital Microfluidic Biochips

DMF Biochip has several limitations, like lack of integrated sensors, fixed droplet size and volume, etc. To resolve these disadvantages of DMF biochip, MEDA-DMF Biochip architecture [4] has been introduced as shown in Figure 2.3. MEDA-DMF architecture is based on the concept of sea-of-micro-electrodes. In this architecture, microelectrodes form the

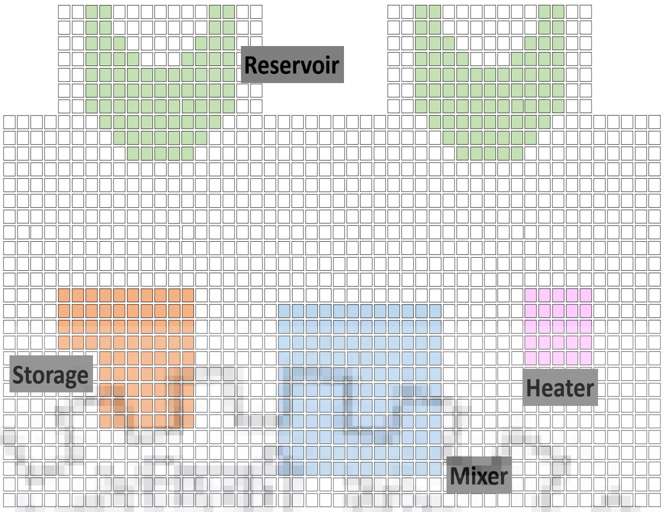


Figure 2.3: Architectural layout of MEDA-DMF biochip.

dynamic micro-component (like mixer) by grouping with each other. Each microelectrode cell consists of an activation circuit, a sensing circuit and a microelectrode. Some of basic differences between traditional DMF and MEDA-DMF biochip are listed here. The fluidic operations are faster on MEDA based biochip. The diagonal movement of droplets is introduced in MEDA-DMF biochip. In MEDA-DMF biochip, complete chip consists of sensors, whereas in DMF biochip, sensors are present on specific area. The flexibility of controlling size, shape and volume of droplet is available in MEDA-DMF biochip. It is not possible in DMF biochip.

2.4 Types of Resources

A DMF biochip can be viewed as a dynamically reconfigurable system consisting of virtual microfluidic modules. If cells in a microfluidic module are busy or faulty, this module can easily be relocated to some another part of the microfluidic array through reconfiguration. The two types of resources are as follows:

2.4.1 Reconfigurable Resources

Reconfigurable resources are those resources which can be reconfigured, i.e., same group of cells in a microfluidic array can be used to perform different functionalities. For example, a group of cells is configured as a mixer and later same group of cells can be considered for a split operation.

2.4.2 Non-reconfigurable Resources

Non-reconfigurable resources are those resources which cannot be reconfigured. They have fixed locations. For example, detectors, heaters and coolers are non-reconfigurable resources and cannot be set at any group of cells.

2.5 Graph and Geometry Concepts for Fluid Routing

A routing algorithm is proposed which is based on computational geometry and graph concepts. The algorithm uses a few algorithms which are discussed in this section.

2.5.1 Convex Hull for a Set of Points

Convex hull [27] of a set of points is defined as the smallest convex polygon that encloses all the points in the set. In Figure 2.4, left figure shows a set of points and right figure shows the convex hull computed for those points. This concept of convex hull is used in finding the shortest path in a space with obstacles. Given source S and destination D, shortest obstacle free path is to found out. The shortest path is always a straight line. If that straight line is not obstacle free, the shortest path would be the one passing right from the corners of the obstacle, i.e., convex hull of S, vertices of intersecting obstacle and D. Two possible paths are visible after computing convex hull. The shorter path is selected to traverse.

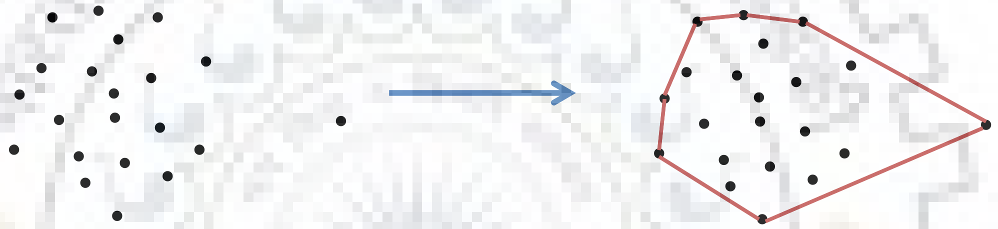
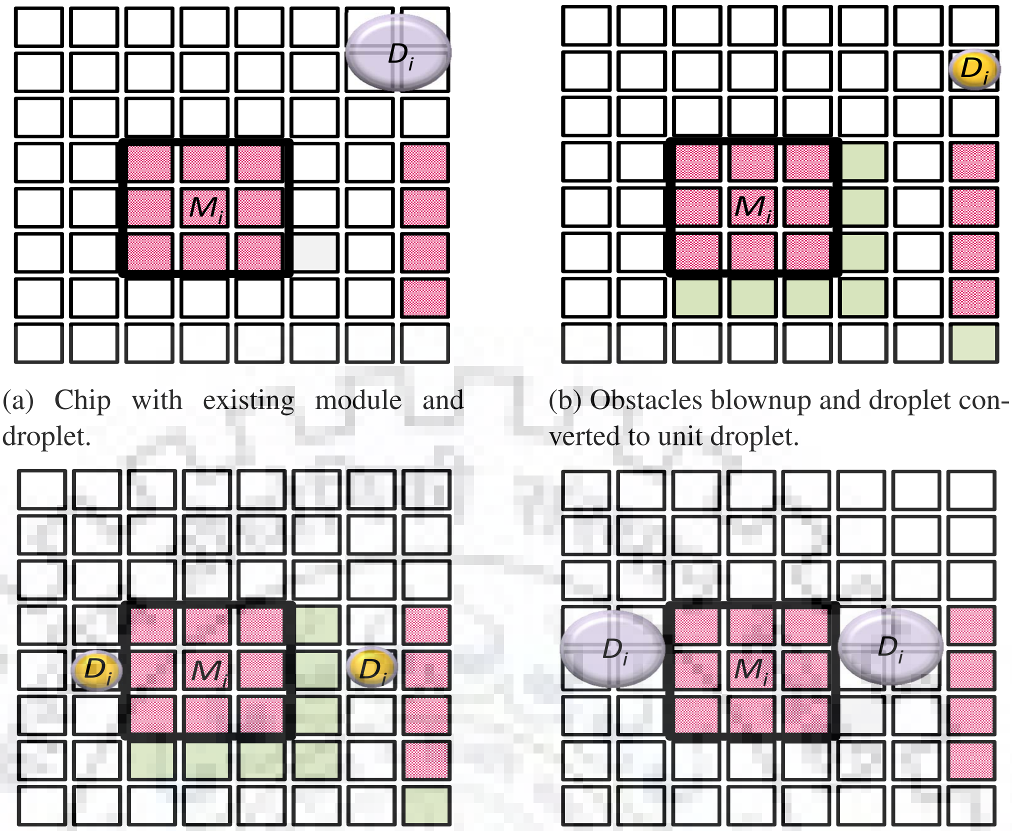


Figure 2.4: Convex hull.

2.5.2 Converting a Rigid Body to a Point Robot

In MEDA-DMF biochip architecture, the droplet size can vary. In the previous works done so far in MEDA-DMF biochip, each time a droplet moves, the algorithm checks if there is sufficient space for the droplet to move. If not, it calculates the space by which droplet can move by changing its shape. This computation of rechecking and recalculating the available space increases the complexity. So, we need to convert a large size droplet to unit size droplet to avoid such complexities. Blowin up concept in *Robot Motion Planning* is used. The chip and obstacles are blown up in such a way that the droplet is considered to be a unit droplet and move freely around the chip. For this, we first select the reference point within the droplet. Each side of the obstacle is blownup by half the size of droplet such that unit droplet location is the reference point of the droplet.

This concept can be well explained with the help of Figure 2.5. Figure 2.5(a) shows existing modules on chip and a droplet Di. Droplet is larger than unit size droplet and so, it needs to be converted to a unit size droplet. The sides of obstacles are blown up according to the reference point selected for the droplet. Figure 2.5(b) shows the droplet converted and obstacles blown up. Figure 2.5(c) shows the movement of the droplet around the chip as unit droplet where the blown up obstacles will act as obstacle for the unit droplet. Figure 2.5(d) shows the same movement when actual droplet moves in real configuration. This



(a) Chip with existing module and droplet.

(b) Obstacles blownup and droplet converted to unit droplet.

(c) Movement of the droplet around the chip as unit droplet.

(d) Same movement of the actual droplet around the chip.

Figure 2.5: Converting a rigid body to a point robot.

way, there is no need to check again and again if enough space is available for the droplet or not.

2.5.3 Dijkstra's Algorithm

In MEDA-DMF biochip, at each configuration, a no of modules may be present. These modules act as obstacles in a configuration. The algorithm of finding a shortest path using convex hull after a few iterations results in a number of possible paths. These possible paths can be merged together to form a graph.

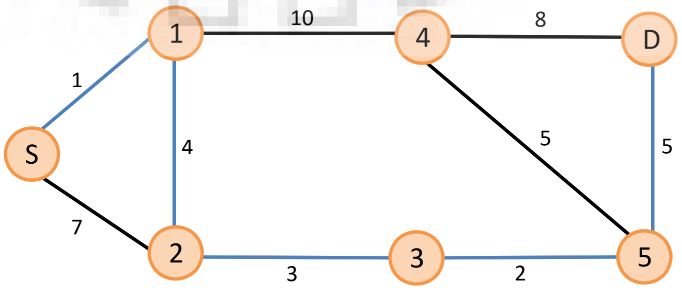


Figure 2.6: Dijkstra's algorithm: shortest path is marked with blue edges in the given graph.

Given a graph with weighted edges, shortest path algorithms can be used to find shortest path. We use Dijkstra's algorithm. This is as shown in Figure 2.6.

2.5.4 Constraint Map for Handling Mobile Droplets

There are two types of obstacles: fixed and mobile. Fixed obstacles can be handled with convex hull. Mobile obstacles for a droplet will be other droplets in the configuration which are moving. To handle these mobile droplets, constraint map [28] is used. It keeps information about which cells are occupied at each time step so that no new droplet occupies those cells at that time step. Figure 2.7 shows the basic structure of constraint map. This constraint map can also output delays and backtrackings.

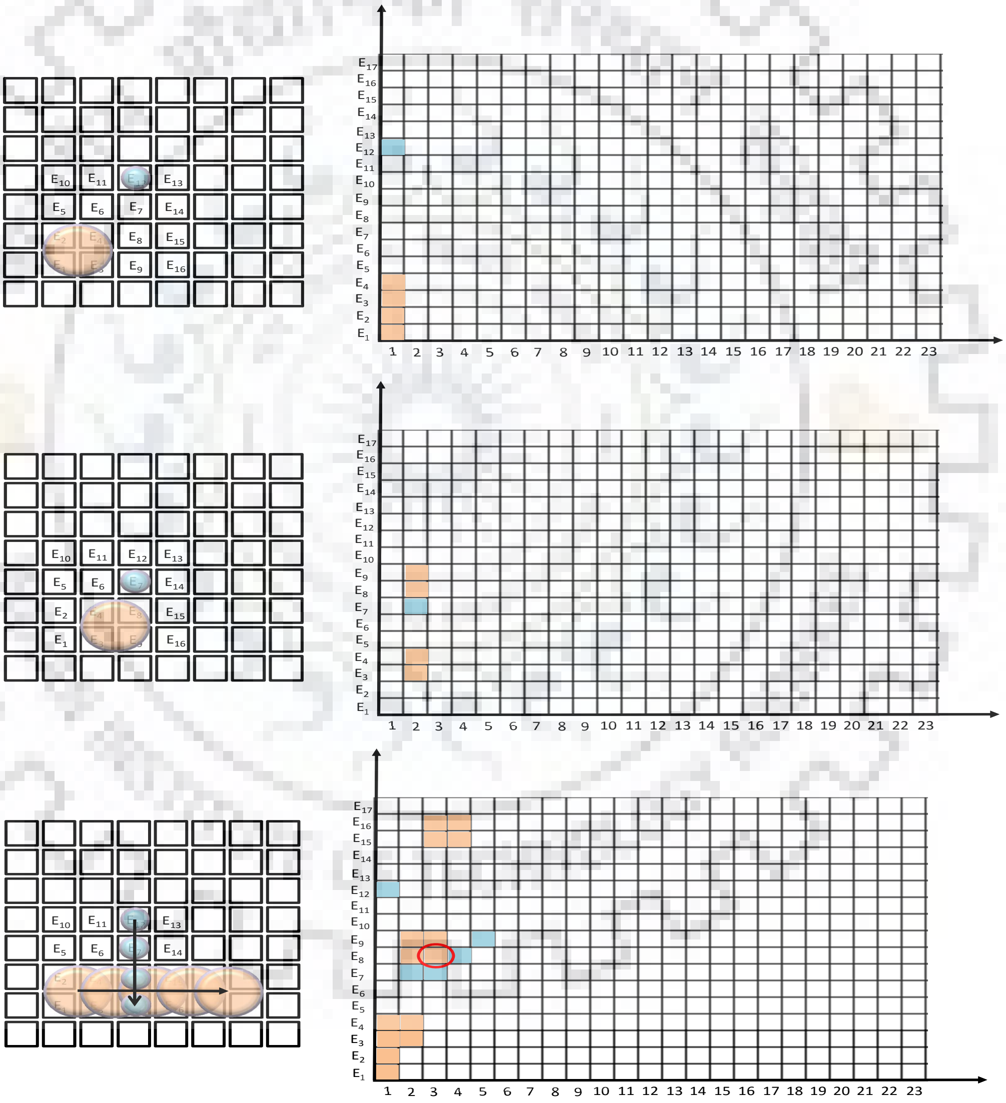


Figure 2.7: Movement of droplets mapped in a constraint map..

2.5.5 Ray Casting Algorithm

Given a polygon and a point, ray casting algorithm finds whether the point is inside or outside a polygon. This algorithm is used in the proposed routing algorithm while removing overlapping of blownup obstacles. The algorithm works as follows. A horizontal line is drawn to the right of the given point and extended to infinity. Count the number of intersections between horizontal line and polygon edges. If the count of intersections is odd, either point is inside the polygon or point lies on some edge of polygon. And if the number of intersections is even, then point lies outside the polygon. This can be seen in Figure 2.8.

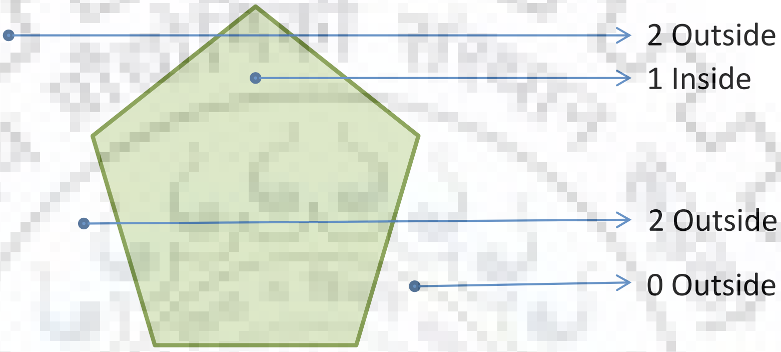


Figure 2.8: Ray casting algorithm.

Chapter 3

LITERATURE SURVEY

3.1 Synthesis of DMF Biochip

A bioassay or bioprotocol is represented in the form of a sequence graph. Nodes are operations and edges define the flow of operation. Synthesis of a DMF biochip has a number of steps which are shown in Figure 3.1. This contains three major steps:

- Scheduling of operations is done which determines the start and stop times of operations.
- Then placement (resource binding) is done for each operation which determines the locations of each module on the microfluidic array.
- Finally, routing of fluids is done which determines the path to be followed by fluids on microfluidic array.

3.1.1 Scheduling of Sequencing Graph for DMF Biochip

List Scheduling Algorithm [1]

- Input: a sequence graph, resource constraints.
- Output: a scheduled graph.
- Method: resources of each kind are assigned to the operations such that number of unfinished and selected operations at a particular time instance do not exceed the number of available resources.

Genetic Algorithm [1]

- Input: a sequence graph, resource constraints.
- Output: a scheduled graph.

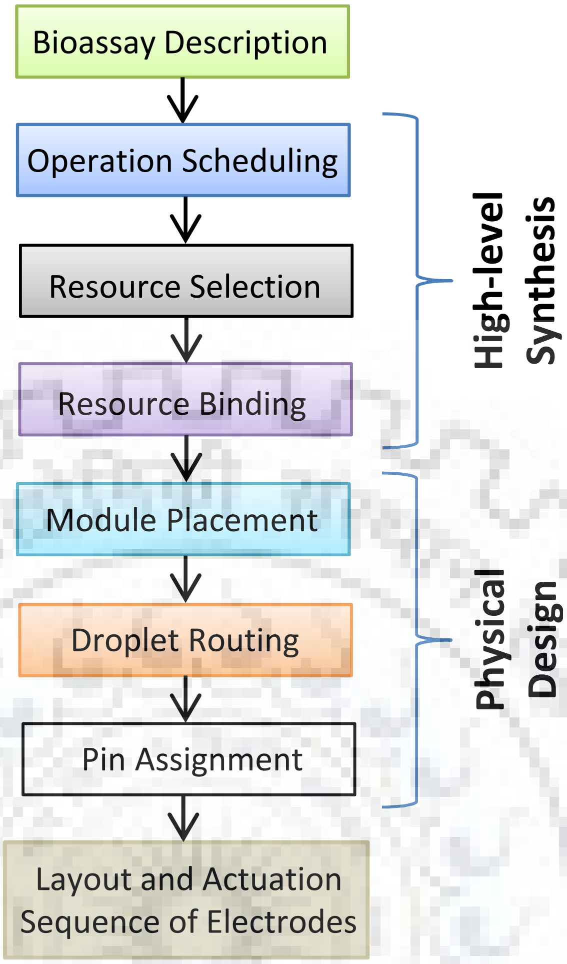


Figure 3.1: Synthesis of DMF biochip.

- Method: scheduling is done based on the fitness value. For each generation, a new population of chromosomes is created and fitness value is recorded. The optimum chromosome with the highest fitness value from the final population after G generations of evolution is selected.

3.1.2 Placement for DMF Biochip

Flow Scan Algorithm [6]

- Input: configuration of chip at an instance.
- Output: complete set of maximum free rectangles at that instance.
- Method: this algorithm scans the chip from bottom to top searching for maximum free rectangles (busy rectangles are mixers or storage modules). From these rectangles, we can assign the best fit rectangle to each new module arriving at that time instance.

Fault Tolerance for DMF Biochips [1]

There are chances that a cell becomes faulty while performing an operation on biochip. The module can be relocated to another part of the array which is non-faulty and unused. Fault Tolerance Index (FTI) is a measure to evaluate the fault tolerance capability of the microfluidic biochip. The value lies between 0 and 1. When FTI=1 means that the chip has good fault tolerance capability, i.e., when any arbitrary cell is determined as faulty, the chip can be reconfigured.

3.1.3 Routing in DMF Biochip

Lee's Algorithm [7]

- Input: chip configuration with blocked and unblocked cells, source and target.
- Output: shortest path between source and target.
- Method: this algorithm uses the principle of BFS. Source is selected and marked as '0'. For each selected cell i (marked value= i), space is explored by marking each adjacent neighbour of cell i as $i+1$ and so on until target cell is reached or no further space can be explored.

Hadlock's Algorithm [7]

- Input: chip configuration with blocked and unblocked cells, source and target.
- Output: shortest path between source and target.
- Method: this algorithm uses A* search method and minimum detour number. The length of path connecting source and target is given by manhattan distance between source and destination plus twice the detour number. The detour number is the number of times the path has turned away from the target.

3.2 Digital Microfluidic Biochip Static Simulator

The UCR Digital Microfluidic Biochip Static Simulator [10] is a tool or repository of algorithms that compile specifications of bioassays onto DMF Biochips. The tool provides complete graphics for input as well as output. It works in three projects: MFSimStaticGUI (for taking input algorithms), MFSimStatic (contains all scheduling, placement and routing algorithms' source code) and DmfbSimVisualizer (for output visualization of chip). Route output of DmfbSimVisualizer is shown in Figure 3.2. We have installed and studied the code of tool to use it further for visualization in MEDA-DMF biochip. Below are some screenshots of the installed tool.

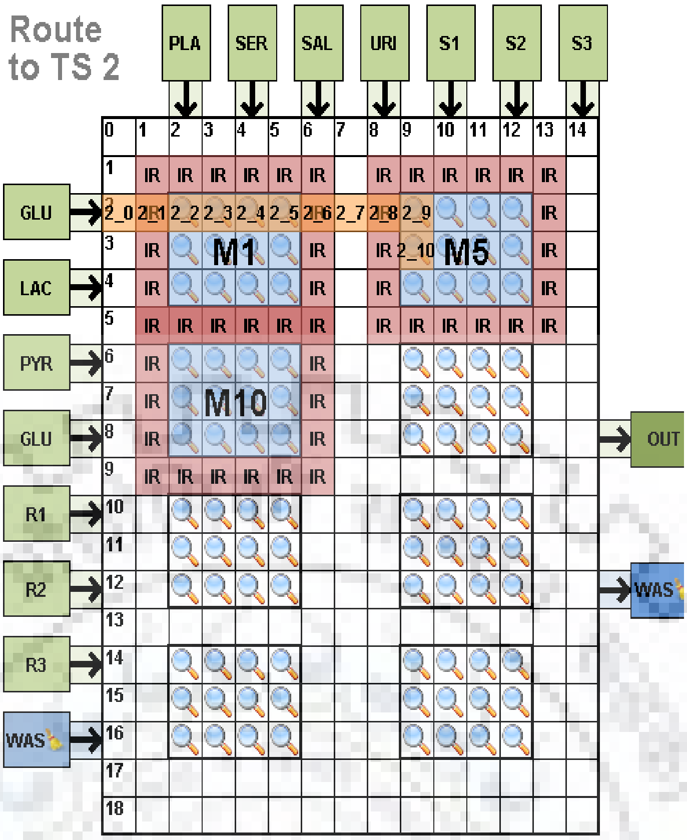


Figure 3.2: Output of DmfbSimVisualizer [10].

3.3 Prior Work in Design Automation for MEDA-DMF Biochip

In the last few years, a huge body of work has been done on MEDA-DMF biochip. For example, different algorithms for error recovery, fault tolerance, security and reliability aware issues have been proposed [11–16]. Sample preparation on microfluidic biochip is another important application, few recent work has been proposed [17–19] for efficient sample preparation which exploits the advantages provided by the MEDA-DMF biochip. Real-time based placement for MEDA-DMF biochip has been proposed in [20], this is the only work on the placement problem of the physical design flow of MEDA-DMF biochip. The first routing algorithm for MEDA-DMF biochip [21] includes the routing of different shape droplets but exclude the shape morphing possibilities. A droplet size aware routing algorithm for MEDA-DMF biochip was proposed in [22] but it does not consider the possibility of diagonal movement of the droplets. Later in [23], an ILP based exact routing on MEDA-DMF biochip was put forward which considers both shape morphing and diagonal movement of droplets. Recently, another negotiation-based routing algorithm [24] and a complete design synthesis approach [25] has been proposed for MEDA-DMF biochip.

3.3.1 Routing Algorithms for MEDA-DMF Biochip

3D-A* Routing Algorithm [8]

- Input: chip configuration with blocked and unblocked cells, set of 'source and target'.
- Output: shortest paths between source and target.
- Method: this uses 3D-A* search algorithm to find path. In this algorithm for MEDA-DMF biochip biochips, droplet routing on an electrode array is considered as routing in 3D space. The x and y coordinates represent the position of electrodes on chip. The z coordinate corresponds to time steps. All the movements are allowed: horizontal, vertical and diagonal. There are a total of 9 positions possible for next time step: $(x+1,y,t+1),(x-1,y,t+1),(x,y+1,t+1),(x,y-1,t+1),(x,y,t+1),(x+1,y+1,t+1),(x+1,y-1,t+1),(x-1,y-1,t+1)$ and $(x-1,y+1,t+1)$. It becomes similar to a 3D path finding problem.

Multi-Level Droplet Routing Algorithm [9]

- Input: chip configuration with blocked and unblocked cells, set of 'source and target'.
- Output: shortest paths between source and target.
- Method: the algorithm consists of 2 stages, top-down uncoarsening followed by bottom-up coarsening. In the uncoarsening stage, a non-splitting reshaping-driven detailed routing algorithm is used. In the coarsening stage, droplets failing above stage are considered. The droplet is split into smaller droplets and routed.

3.3.2 Prior Work for Synthesis in MEDA-DMF Biochip

In 2017, Zipeng Li proposed a synthesis approach [25] for MEDA-DMF biochip architecture where he targets operation scheduling, module placement, routing of droplets of various sizes, and diagonal movement of droplets in a two-dimensional array. In this approach, scheduler calls placer and after deciding space on chip for operation module, placer calls router. The detail of algorithm is discussed in Section 4.1.2.

Chapter 4

RESEARCH GAPS AND PROBLEM STATEMENT

4.1 Research Gaps

4.1.1 Specification of Bioassay, Architecture and Module Library for MEDA-DMF Biochip

There are a number of operations performed by MEDA biochips, for example, dispense, mix, heat, split, detect. All these operations will take different time for different volume droplets. The situation gets worse when reconfigurability of different resources is considered. For example, each dispenser is reserved for a particular fluid whereas output ports can be used for any droplet. A small-sized heater cannot accomodate droplets of all volumes whereas a mixer or output port can. This variability in functionality of different operations increases the need of a well-defined library and architecture. There are no such existing files. Hence, these files is proposed in Chapter 5 along with bioassay file.

4.1.2 A New Synthesis Approach for MEDA-DMF Biochip

Synthesis in MEDA-DMF biochip is different from that of DMF biochip because of the following reasons:

- **Droplets may vary in size, volume and shape**: A droplet can change its shape and size while traversing on chip which is not possible in DMF biochip. Droplets in MEDA-DMF biochip can be of different volumes whereas it is fixed in DMF biochip.

- **Droplet dispensing time is no longer fixed**: Droplets in DMF biochip have equal volume and so the time to dispense fluid out of reservoir is same for all the droplets. Unlikely in MEDA-DMF biochip, droplets can be of different volumes, their dispense time will be different.

- **Diagonal movement of droplets**: Droplet in MEDA-DMF biochip creates effective line of contact due to their size so that it can be pulled diagonally. There is no such contact in DMF biochip.

- **Droplet routing time considered**: Routing time is considered negligible to the mixing time in DMF biochip. Hence, it is ignored. But it is important to consider routing time in MEDA-DMF biochip due to SAR mixer which makes mixing time comparable to routing time.

Extensive research work has been done on the high-level and architectural-level synthesis of DMF biochip whereas very few work can be found in the literature for the same problems for MEDA-DMF biochip. Zipeng Li et.al [25] proposed a synthesis approach for MEDA-DMF biochip. In the current literature, it is the only work which deals with the design automation for complete synthesis, i.e., scheduling, placement and routing for MEDA-DMF biochip. He proposed an approach where scheduling is done considering placement. And route is calculated after placement of a module.

In this approach, an operation is scheduled by first calculating operation time window using arrival time and execution time of the operation and then calling placer to find out space on chip which is free for that operation time window. A module is placed on chip after finding out all the existing modules for that time window. Later, routing time from parents of that operation is calculated and added to the arrival time of that operation. As a result, start time and end time of the operation changes which may lead to failure of placement.

This failure can be explained with the help of an example. A mixing operation O arrives at TS = 25 and execution time of the operation is 5 units. For scheduling, placer is called to find out module space which is free for TS = 25 to TS = 25+5 = 30. After placement is successful, routing time is calculated from Parent(O) (parent operation(s) of operation O) to the operation O itself. For instance, routing time calculated is 8 units. This routing time changes the operation time window as the time taken to reach from parent operation is added. Hence, the module will start working at TS = 25 + 8 and end at TS = 25+8+5. This change in operation time window may lead to failure as the selected space may not be free for the new time window. There is a possibility that some other module is working on the selected space.

This concludes that routing time should be considered with placement simultaneously. For each operation, validity of the space should be checked after routing time is calculated for the candidate free space.

4.2 Problem Statement

This dissertation tries to achieve a single-flow or one-pass design synthesis process for MEDA-DMF biochip. In the conventional synthesis flow, we do not have any choice to re-plan the placement while the routing phase. So, we consider the placement and the routing phase simultaneously here in proposed synthesis. A new routing algorithm is also proposed which takes care of both any-direction flow and shape morphing properties provided by a MEDA-DMF biochip. Also, considering the lack of well-defined libraries, architectures and bioassay files, we propose the structure of library file, architecture file and bioassay file. We also propose benchmarks for MEDA-DMF biochip architecture.

Chapter 5

PROPOSED MODELLING FOR DESIGN AUTOMATION OF MEDA-DMF BIOCHIP

Modelling here refers to deciding the structures of different input files. These input files are bioassay files, architecture files and library files. There are no such existing files as mentioned in research gaps. We need to study the different types of fluidic operations to decide the specifications of these files.

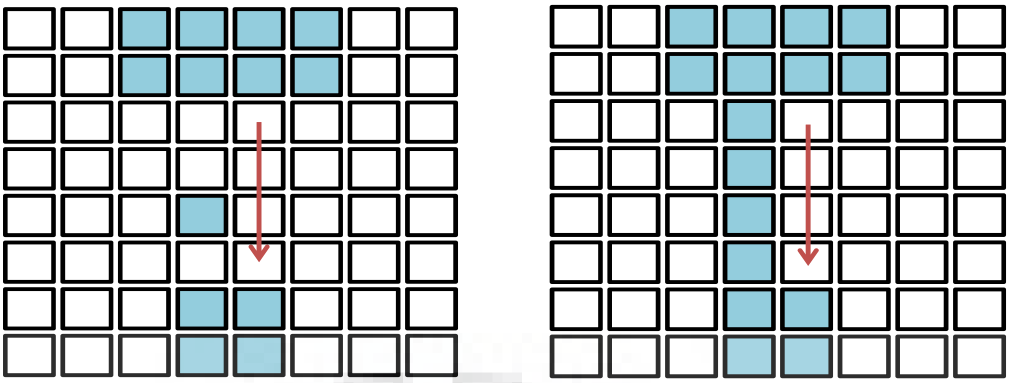
5.1 Types of Fluidic Operations in MEDA-DMF Biochip

MEDA-DMF biochip is able to perform all the fundamental EWOD microfluidic operations successfully. In addition, it performs many advanced microfluidic operations which other architectures are unable to perform. Following are the operations performed by MEDA-DMF biochip:

5.1.1 Fluid Dispense from Reservoirs

When a fluid is dispensed from reservoir to the chip to create a droplet, it is called a dispense operation. Droplets can be created by two methods. One method for creating the droplet is droplet aliquots. At first, create smaller droplets from reservoir then collect smaller droplets to create bigger droplet by activating more micro-electrodes. Droplet aliquots is used for measuring the volume of droplets. Another way of creating droplet is to form bridge between destination location of droplet and reservoir using microelectrodes. After that activated destination and bridge microelectrodes will cause fluid to flow from reservoir to destination electrode. When fluid reaches destination electrode then bridge microelectrodes get deactivated. The two methods are as shown in Figure 5.1.

Each reservoir is reserved for a particular fluid. So, the reservoirs are non-reconfigurable resources. Hence, dispense operation of a fluid will use a particular reservoir.



(a) Droplet showing aliquots.

(b) Droplet creation using bridge.

Figure 5.1: Converting a rigid body to a point robot.

5.1.2 Fluid Output and Waste through Ports

Output operation is done using output ports. No output port is reserved for a particular fluid. Any droplet can leave from any port. Hence, for any output opearation, a free output port can be selected. Similar to the output port, waste port can also be used for any fluid. Generally, there is only one waste port on every chip and each waste operation uses the same port to output waste. Difference between the two ports is that output port is used to collect the useful fluids prepared from synthesis but waste port leaves out useless fluids only.

5.1.3 Mix of Multiple Fluids

Mix operation in MEDA-DMF biochip architecture is different from that of DMF biochip architecture. MEDA-DMF biochip uses the SAR lamination mixing technique. SAR consists 3 major steps: fluid element splitting, rearrangement and recombination as shown in Figure 5.2. Typically, it takes three cycles of the SAR lamination mixing to mix the droplet well. This technique of mixing two fluids takes relatively less time than mixing in DMF biochip. In DMF biochip, routing time is ignored as the mixing time is considered relatively larger than any routing time. But as the mixing time is reduced in MEDA-DMF biochip, the two time durations become comparable and routing time needs to be considered. A mix operation is carried out on a group of cells. These group of cells is decided based on the mixer size and the availability of cells on chip. A mixer size is decided according to the droplets' sizes to be mixed. This operation is reconfigurable as the same group of cells can be used for some other operation. So, for any mix operation, free space according to the mixer size requirement is found and assigned as mixer.

5.1.4 Splitting a Droplet into Two Droplets

When a droplet is to be divided into two equal halves, it is called a split operation. While scheduling, these operations are operated in the similar way as they are reconfigurable and any free space on chip can be used to execute this operation. An important point to note here is that the operation will split into two parts only.

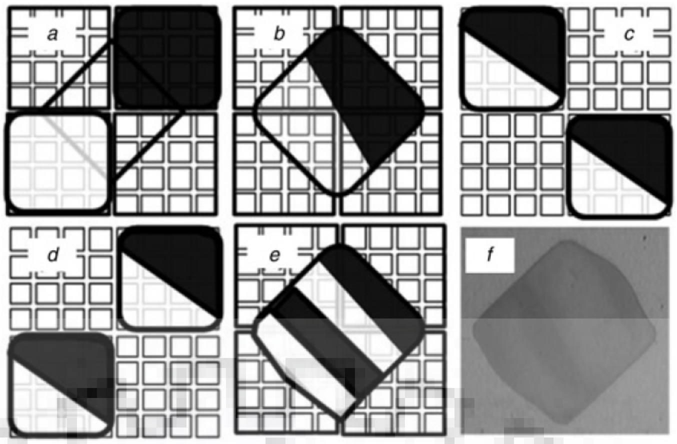


Figure 5.2: Droplet mixing SAR technique [26]

5.1.5 Separating Fluid from a Droplet

When a small portion of the droplet is to be separated from a bigger droplet, it is called a separate operation. The difference between a split and separate operation is that the former divides the droplet in two equal parts and later separates a small part from the droplet. A separate operation will have two children in this work for easy convention. This part is explained in Section 5.2 in much detail.

5.1.6 Concentration Detection, Heating and Cooling of Fluid

CDetect (concentration detect), heat and cold operation detects, heats and cools the droplet respectively. These operations perform similar scheduling. The resources of these operations are present on the chip at some locations. More than one resource for an operation is possible here. For example, a chip has two heaters, both of different sizes. A droplet of large size can be accomodated on the larger heater only. So, the droplet will wait for the larger heater only. For these operations, the resource is decided based on the volume of the droplet so that this droplet can be accomodated in it. These operations are non-reconfigurable.

5.1.7 Volume Detection of Fluid

MEDA-DMF biochip architecture has an advantage that it has sensors below each microelectrode, i.e., it can measure volume anywhere on chip. These operations are reconfigurable. A VDetect (volume detect) operation can be executed on the free space on chip. For every such operation, placement is done using different algorithms and free space is computed with respect to droplet size.

5.2 Input Files for Synthesis of MEDA-DMF Biochip

As mentioned in Chapter 4, there is a need of well-defined input files. These input files are bioassay file, architecture file and library file. We propose benchmarks for MEDA-DMF biochip. Here is the list of ten testcases selected for this work:

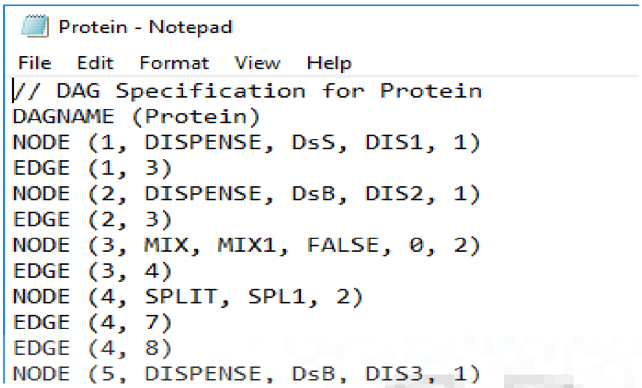
1. InVitro1
2. FacDA
3. MRCM
4. MTCS
5. PCR-Mix
6. Protein
7. WSPM
8. Synthetic1
9. Synthetic2
10. Synthetic3

The last three bioassays: Synthetic1, Synthetic2 and Synthetic3 are self made so that all the operations can be verified. Synthetic1 is the only bioassay which contains all the ten operations.

5.2.1 Bioassay File for Sequencing Graph

Figure 5.3 shows the screenshot of the bioassay file prepared for this work. A bioassay file contains sequence of operations to be performed for a bioassay. Each operation has a different syntax. There are different types of tags defined in this file:

- DAGNAME: this tag sets the name of the bioassay.
- TIME_UNIT: this tag sets the unit of time as microsecond, millisecond and second.
- VOLUME_UNIT: this tag sets the unit of volume as microlitre, millilitre and picolitre.
- TEMPERATURE_UNIT: this tag sets the temperture unit as celcius, kelvin.
- MICROELECTRODES_PER_UNIT_VOLUME: this tag describes the number of microelectrodes on which unit volume of fluid can sit.
- NODE: it defines an operation. Syntax of each operation is described later in the same section.



```
// DAG Specification for Protein
DAGNAME (Protein)
NODE (1, DISPENSE, DsS, DIS1, 1)
EDGE (1, 3)
NODE (2, DISPENSE, DsB, DIS2, 1)
EDGE (2, 3)
NODE (3, MIX, MIX1, FALSE, 0, 2)
EDGE (3, 4)
NODE (4, SPLIT, SPL1, 2)
EDGE (4, 7)
EDGE (4, 8)
NODE (5, DISPENSE, DsB, DIS3, 1)
```

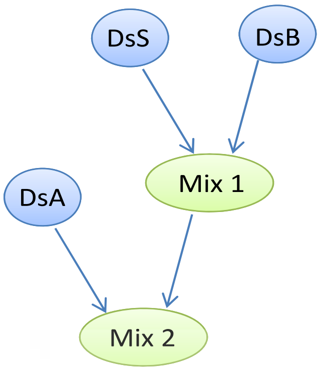


Figure 5.3: Bioassay file and DAG.

- EDGE: edge connects two nodes, defining the sequence or dependence of operations. The two parameters of edge are the node IDs of the two connecting nodes.

The syntax of different operations is described as follows:

- NODE (1, DISPENSE, plasma, DIS1, 1)
  This has 5 parameters. These parameters can be described as (nodeID, operation type, fluid name, operationID, volumne of the fluid).

- NODE (3, MIX, MIX1, FALSE, 0, 2)
  This has 6 parameters. These parameters can be described as (nodeID, operation type, operationID, true if child operation execution constraint is present else false, value of the constraint if previous parameter is true else zero, volume of the fluid).
  Child Operation Execution Constraint: there are a few operations like mix, heat, cool which results in a fluid which should be used within some restricted time. For this, a boolean variable in fourth parameter is set to true which tells that the output fluid of this operation is valid for a particular time period and fifth parameter gives the value of that time duration. If there is no such restriction, boolean variable is set to false.

- NODE (40, OUTPUT, OUT1, 2)
  This has 4 parameters. These parameters can be described as (nodeID, operation type, operationID, volume of the fluid).

- NODE (24, WASTE, WST1, 7)
  This has 4 parameters. These parameters can be described as (nodeID, operation type, operationID, volume of the fluid).

- NODE (9, SEPARATE, 6, SEP3, 8)
  This has 5 parameters. These parameters can be described as (nodeID, operation type, volume to be separated, operationID, total volume of the fluid).
  In separation node, a volume can be divided into a number of parts. This could be done by using ratio. For example, (1:2:3:5) is the ratio defined to separate a 10 unit volume into parts. But by using the concept of ratios, it is difficult for the user to explain that which part belongs to which child. For this, a mapping of each ratio

component to child node is to be done which becomes difficult and complex for the user. So, an alternative is proposed here. Separate node will always have two children and the volume to be separated will always belong to the child node with higher ID. We differ the two children with the help of node IDs here. We cannot process them as left and right child node because to represent a DAG, *Graphiz* library is used which will not put the two nodes in same manner everytime.

- NODE (11, SPLIT, SPL1, 8)
  This has 4 parameters. These parameters can be described as (nodeID, operation type, operationID, total volume of the fluid).
  A split operation always divides total volume into two equal parts. Hence, there is no such confusion about child nodes and there is no need to mention part to be separated.

- NODE (13, HEAT, HET1, 30, TRUE, 3, 8)
  This has 7 parameters. These parameters can be described as (nodeID, operation type, operationID, temperature, true if child operation execution constraint is present else false, value of the constraint if previous parameter is true else zero, total volume of the fluid).

- NODE (16, COOL, COL1, 5, TRUE, 4, 7)
  This has 7 parameters. These parameters can be described as (nodeID, operation type, operationID, temperature, true if child operation execution constraint is present else false, value of the constraint if previous parameter is true else zero, total volume of the fluid).

- NODE (26, CDETECT, CDT1, 14)
  This has 4 parameters. These parameters can be described as (nodeID, operation type, operationID, volume of the fluid).

- NODE (30, VDETECT, VDT1, 21)
  This has 4 parameters. These parameters can be described as (nodeID, operation type, operationID, volume of the fluid).

5.2.2 Architecture File for MEDA-DMF Biochip

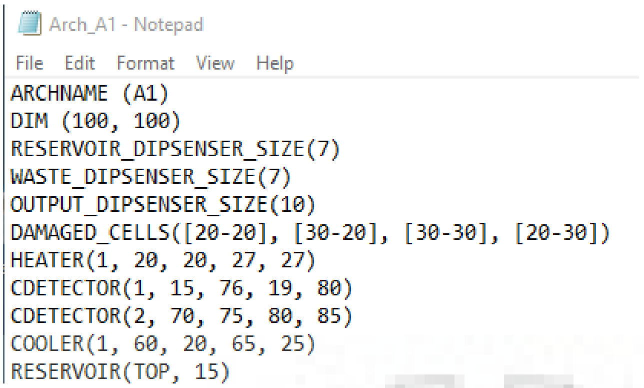
A biochip may have different architectures. An architecture is defined by chip size, number of components, size of dispensers, etc. Here is a well defined structure proposed and used for this work as shown in Figure 5.4. There are different types of tags used in this file:

- ARCHNAME: this tag sets the name of architecture.

- DIM: this tag defines the size of chip.

- RESERVOIR_DISPENSER_SIZE: this tag defines the size of dispenser outside reservoir.

- WASTE_DISPENSER_SIZE: this tag defines the size of dispenser outside waste port.



```
ARCHNAME (A1)
DIM (100, 100)
RESERVOIR_DIPSENSER_SIZE(7)
WASTE_DIPSENSER_SIZE(7)
OUTPUT_DIPSENSER_SIZE(10)
DAMAGED_CELLS([20-20], [30-20], [30-30], [20-30])
HEATER(1, 20, 20, 27, 27)
CDETECTOR(1, 15, 76, 19, 80)
CDETECTOR(2, 70, 75, 80, 85)
COOLER(1, 60, 20, 65, 25)
RESERVOIR(TOP, 15)
```

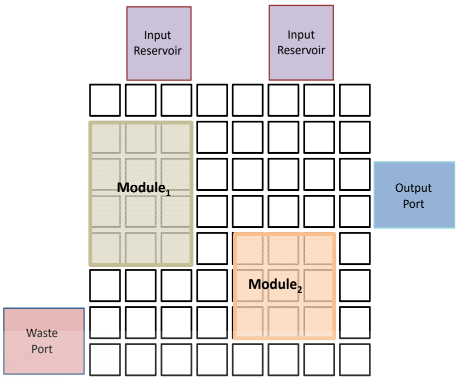


Figure 5.4: Architecture file and geometry of chip.

- OUTPUT_DISPENSER_SIZE: this tag defines the size of dispenser outside output port.

- DAMAGED_CELLS: there is a possibilty where few microelectrodes get damaged. These electrodes cannot be used during synthesis and need to act blocked.

- HEATER: this tag resembles the presence of a heater with its location on chip. For example, HEATER(1, 20, 20, 27, 27) is defined as (heaterID, topLeftX, topLeftY, bottomRightX, bottomRightY).

- CDETECTOR: this tag resembles the presence of a concentration detector with its location on chip. For example, CDETECTOR(1, 20, 20, 27, 27) is defined as (cDetectorID, topLeftX, topLeftY, bottomRightX, bottomRightY).

- COOLER: this tag resembles the presence of a cooler with its location on chip. For example, COOLER(1, 20, 20, 27, 27) is defined as (coolerID, topLeftX, topLeftY, bottomRightX, bottomRightY).

- RESERVOIR: this tag resembles the presence of a reservoir with its location on chip. For example, RESERVOIR(TOP, 15) is defined as (direction, coordinate). The coordinate field will be x coordinate when direction is either TOP or BOTTOM and y coordinate when direction is either LEFT or RIGHT.

- WASTE PORT: this tag resembles the presence of a waste port with its location on chip. For example, WASTE(TOP, 15) is defined as (direction, coordinate). The coordinate field will be x coordinate when direction is either TOP or BOTTOM and y coordinate when direction is either LEFT or RIGHT.

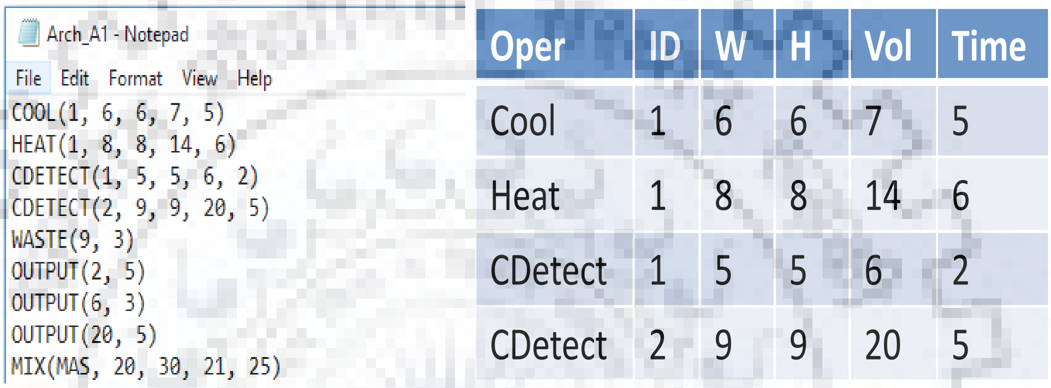
- OUTPUT PORT: this tag resembles the presence of a output port with its location on chip. For example, OUTPUT(LEFT, 55) is defined as (direction, coordinate). The coordinate field will be x coordinate when direction is either TOP or BOTTOM and y coordinate when direction is either LEFT or RIGHT.

- FREQUENCY: this tag defines frequency, i.e., number of cycles in unit time.

- TIMESTEP: this tag defines timestep which is the duration o a timestep in seconds. For example, TIMESTEP(0.5) tells the duration of a timestep is 0.5 seconds, i.e., one second can be divided into two timesteps.

5.2.3 Module Library for MEDA-DMF Biochip

Library file as shown in Figure 5.5 is mainly used to map different operations to their execution time. Different operations mean droplets with different sizes and volumes and with different operation types. For each type of operation, the listed operation should be in increasing order of droplet's volume. Execution time defined is in seconds. The syntax of each operation is different and is defined as follows:



```
COOL(1, 6, 6, 7, 5)
HEAT(1, 8, 8, 14, 6)
CDETECT(1, 5, 5, 6, 2)
CDETECT(2, 9, 9, 20, 5)
WASTE(9, 3)
OUTPUT(2, 5)
OUTPUT(6, 3)
OUTPUT(20, 5)
MIX(MAS, 20, 30, 21, 25)
```

Oper	ID	W	H	Vol	Time
Cool	1	6	6	7	5
Heat	1	8	8	14	6
CDetect	1	5	5	6	2
CDetect	2	9	9	20	5

Figure 5.5: Library file and table.

- COOL(1, 6, 6, 7, 5)
  Given volume of a droplet, one can find out the dimension of a component and the execution time. This syntax can be defined as COOL(coolerID in architecture file, width of the module, height of the module, volume of the fluid, execution time).

- HEAT(1, 3, 3, 5, 2)
  Given volume of a droplet, one can find out the dimension of a component and the execution time. This syntax can be defined as HEAT(heaterID in architecture file, width of the module, height of the module, volume of the fluid, execution time).

- CDETECT(2, 5, 5, 7, 10)
  Given volume of a droplet, one can find out the dimension of a component and the execution time. This syntax can be defined as CDETECT(cDetectorID in architecture file, width of the module, height of the module, volume of the fluid, execution time).

- SEPARATE(9, 9, 5, 1)
  Given volume of a droplet, one can find out the dimension of a component and the execution time. This syntax can be defined as SEPARATE(width of the module, height of the module, volume of the fluid, execution time).

- SPLIT(6, 6, 7, 6)
  Given volume of a droplet, one can find out the dimension of a component and the

execution time. This syntax can be defined as SPLIT(width of the module, height of the module, volume of the fluid, execution time).

- VDETECT(4, 4, 3, 2)
  Given volume of a droplet, one can find out the dimension of a component and the execution time. This syntax can be defined as VDETECT(width of the module, height of the module, volume of the fluid, execution time).

- WASTE(1, 3)
  Given volume of a droplet, one can find out the execution time. This syntax can be defined as WASTE(volume of the fluid, execution time).

- OUTPUT(2, 4)
  Given volume of a droplet, one can find out the execution time. This syntax can be defined as OUTPUT(volume of the fluid, execution time).

- DISPENSE(5, 8)
  Given volume of a droplet, one can find out the execution time. This syntax can be defined as DISPENSE(volume of the fluid, execution time).

- MIX(SAR, 4, 4, 3, 2)
  Given volume of a droplet, one can find out the dimension of a component and the execution time. This syntax can be defined as MIX(type of mixer, width of the module, height of the module, volume of the fluid, execution time).

Chapter 6

PROPOSED SYNTHESIS APPROACH FOR MEDA-DMF BIOCHIP

As discussed in Chapter 4, a new synthesis approach is needed which considers the placement and routing phase simultaneously. Synthesis of a MEDA-DMF biochip consists of three major steps which are discussed in detail in below subsections:

1. Scheduling of operations
2. Placement of modules
3. Routing of droplets

6.1 Scheduling of Sequencing Graph for MEDA-DMF Biochip Synthesis

Scheduling is one of the important steps in synthesis of a bioassay. It assigns the start and end time of each operation. As mentioned earlier, placement and routing will be done simultaneously.

6.1.1 Priority Generator

Priority Generator sets the arrival time of each operation. Before starting scheduling, the arrival time of each operation should be decided from the bioassay. Here, we assign the arrival time level wise in a graph. Base level is decided to be one with leaf nodes, i.e., output and waste nodes. So, waste and output nodes are always in the last level and rest of the operations are assigned levels by adding one to their children level. This way, we assign levels to each operation. Input to the scheduler starts from the minimum level and if any operation waits for next time event, it means it is added to the next level.

6.1.2 Proposed Scheduler for MEDA-DMF Biochip

Algorithm 1 describes pseudo code of the proposed scheduler. At first, all the operations are assigned arrival time using priority generator. Then each operation is scheduled by checking successful placement and routing. There are two types of operations. For non-recofigurable operations, we check if corresponding resource is available after droplet reaches the resource. If conditions allow scheduling, the operation is scheduled. For reconfigurable operations, we first find free space on the chip and then check if it will be available after reaching there.

Algorithm 1: Proposed Scheduler

```
Input  : A bioassay file, architecture file and library file corresponding to
         architecture
Output: Scheduled operations of the bioassay file
Set the arrival time of each operation using Priority Generator;
for each operation in priority list do
    if operation is a non-reconfigurable operation then
        Get volume of the droplet from bioassay file and find corresponding resource
          and execution time for the operation from architecture and library file;
        Router /* Calculate routing time to reach the resource and then find if
          resource is available from that time for execution time */
        if resource is available and router is successful then
            Schedule the operation and add routing time to the edges;
        else
            Wait for the next time event;
    if operation is a reconfigurable operation then
        Get volume of the droplet from the bioassay file and find corresponding
          resource location and execution time for the operation from architecture and
          library file;
        Placer /* find free rectangles on the chip */
        for each free rectangle do
            Call Router /* Calculate routing time to reach the resource location and
              then find if resource microelectrodes are available from that time for
              execution time */
            if Placer and Router both are successful then
                Store the result for this free rectangle;
        if any free rectangle was possible then
            Schedule the operation and add routing time to the edges with optimal
              free rectangle;
        else
            Wait for the next time event;
```

6.2 Module Placement for MEDA-DMF Biochip Synthesis

Placement is another important step in synthesis of a biochip. In this work, there is a special case of damaged cells. This leads to the generation of orthogonal obstacles as shown in Figure 6.1. In placement phase, we find free space to assign incoming modules. In DMF biochip, there are a number of algorithms for placement phase but all of them consider the existing modules (obstacles) as rectangles. So, to manage these orthogonal obstacles, an extra step is added to Flow Scan algorithm for DMF biochip.

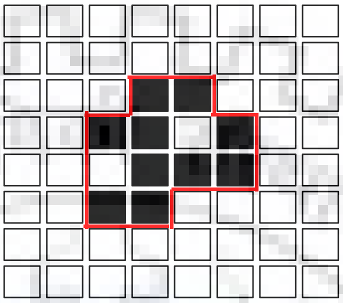


Figure 6.1: Black microelectrodes represent damaged cells.

Break the orthogonal obstacles vertically and treat them as separate rectangular obstacles while chip scanning. Then use the regular Flow Scan algorithm to get free rectangles. The basic working of placement phase in this work can be explained with the help of Figure 6.2.

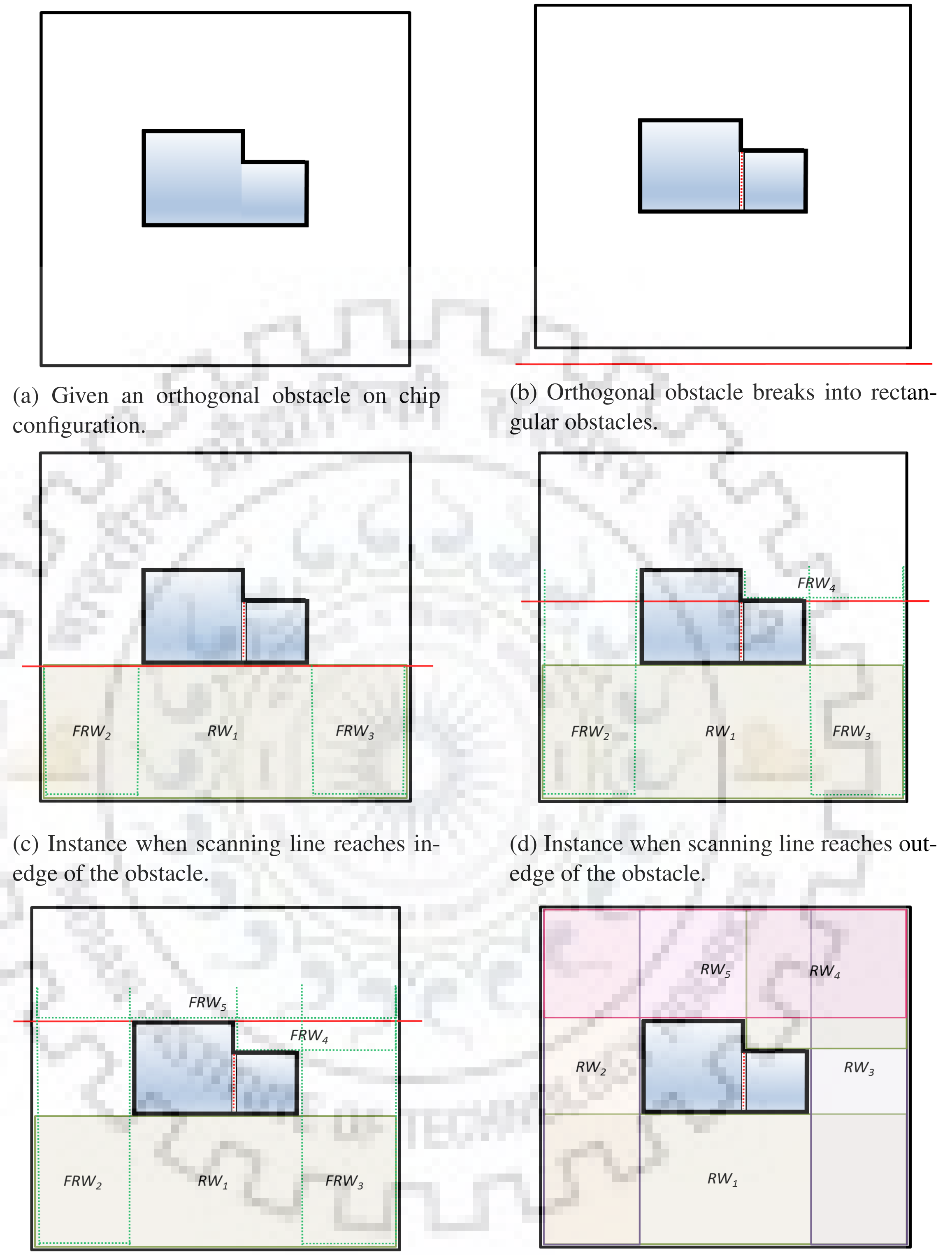
Figure 6.2(a) shows the presence of an orthogonal obstacle. Figure 6.2(b) shows breaking of the orthogonal obstacles into rectangular obstacles. Figure 6.2(c) shows the instance when scanning line reaches inedge of the obstacles. Figure 6.2(d) shows the instance when scanning line reaches one of the outedges of the obstacle. Figure 6.2(e) shows the instance when scanning line reaches the other outedge of the obstacle. Figure 6.2(f) shows all the free rectangles after scanning whole chip.

6.3 Fluid Routing in MEDA-DMF BIOCHIP Synthesis

In DMF biochip, routing time is ignored but here, it needs to be considered. Hence, routing becomes another important step in synthesis. It finds the exact cell to cell traversal of the droplet from source microelectrode to destination microelectrode. A new routing algorithm for MEDA-DMF biochip is proposed in this chapter.

6.3.1 Configuration

A configuration or subproblem here defines the locations of all the mixer, heater, split, detector modules that are present at that time instance. So, given a configuration means all the modules that are present on chip for that time instance are given and they are considered as obstacles while calculating route from source to destination.



(a) Given an orthogonal obstacle on chip configuration.

(b) Orthogonal obstacle breaks into rectangular obstacles.

(c) Instance when scanning line reaches inedge of the obstacle.

(d) Instance when scanning line reaches outedge of the obstacle.

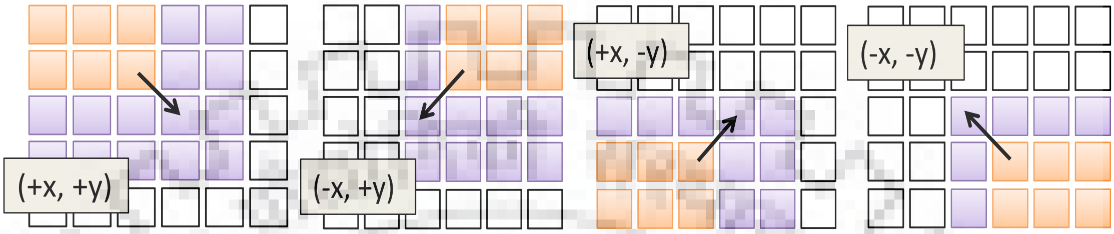
(e) Instance when scanning line reaches another outedge of the obstacle.

(f) All free rectangles are marked as RW after scanning.

Figure 6.2: Flow scan algorithm for orthogonal obstacles.

6.3.2 Computing Blowup Direction and Size

The orthogonal obstacles is structured in form a vector of microelectrodes starting from top left vertex and traversing in clockwise direction. Blownup orthogonal obstacles will have the same structure with the modifications in x and y coordinate. These modifications are made according to the vertex type: top left, top right, bottom left, bottom right which can also be considered as the direction in which a vertex is to be blown up. This can be easily understood from Figure 6.3.



(a) Bottom right vertex. (b) Bottom left vertex. (c) Top right vertex. (d) Top left vertex.

Figure 6.3: Blowup direction and size for different types of vertices.

Each vertex in orthogonal obstacle will be one of the four vertex types as shown in Figure 6.3. Identifying the type of each vertex is another task. The first vertex will always be top left and the direction of every next vertex is calculated based on the reduction or accession of x or y coordinate.

Blowup size means the amount of cells by which the obstacle is to be blown up, i.e., the amount of reduction or accession to be made in x or y coordinate. The amount of blowup depends on the size of droplet and can be computed with the help of Eqn. (6.1 - 6.4).

$$leftBlowCells = \lceil dropletSize/2 \rceil - 1 \tag{6.1}$$

$$rightBlowCells = dropletSize - leftBlowCells - 1 \tag{6.2}$$

$$topBlowCells = \lceil dropletSize/2 \rceil - 1 \tag{6.3}$$

$$bottomBlowCells = dropletSize - topBlowCells - 1 \tag{6.4}$$

where $leftBlowCells$, $rightBlowCells$, $topBlowCells$ and $bottomBlowCells$ are the no of cells that should be blown up in left, right, top and bottom direction respectively. $dropletSize$ is the size of the droplet in terms of cells.

6.3.3 Removing Overlapping of Blownup Obstacles

Whenever we talk of blowing up an obstacle, there is a possibility of overlapping of obstacles as shown in Figure 6.4. This overlapping means that the droplet cannot traverse through that overlapped region in its original shape. Here, shape morphing of a droplet is done.

To remove overlapping and compute the size of droplet at each cell of new generated path, we use Eqn. (6.5 - 6.8).

$$dropletSizeThatCanTraverse = dropletSize - 1 - widthOfOverlappedRegion \tag{6.5}$$

$$middlePathX = leftObstacle'sX - (rightBlowCells - \lceil (dropletSizeThatCanTraverse + 1)/2 \rceil) \tag{6.6}$$

$$leftObstacle'sX = middlePathX - 1 \tag{6.7}$$

$$rightObstacle'sX = middlePathX + 1 \tag{6.8}$$

where $dropletSizeThatCanTraverse$ is the size of the droplet that can pass through the overlapped region, $dropletSize$ is the size of the droplet in terms of cells, $widthOfOverlappedRegion$ is the number of cells along x axis in the overlapped region, $middlePathX$ is the x coordinate of the middle path found out, $leftObstacle'sX$ is the x coordinate of left obstacle's overlapping edge, $rightObstacle'sX$ is the x coordinate of right obstacle's overlapping edge, $rightBlowCells$ computed from Eqn. (6.2).

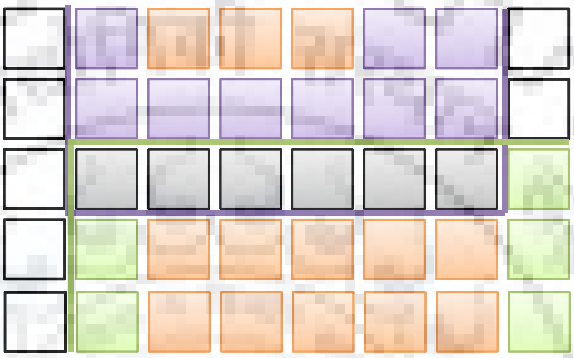


Figure 6.4: Overlapping of blownup obstacles

6.3.4 Duration of Time Step

In MEDA-DMF biochip, there is an advantage that a droplet can make diagonal movement. Hence, in this achitecture, a droplet can make manhattan (horizontal and vertical) and diagonal movement both. The two movements cover different distances and hence, take different times. This leads to a problem of clock cycle, i.e., to decide the number of cycles each movement will take.

We propose a solution to this problem via approximation. 1.44 units distance can be approximated as 1.5 units. So, if we set the clock such that 0.5 unit distance is covered in a cycle then manhattan movement will take two cycles and a diagonal movement will take three cycles.

6.4 Proposed Router for MEDA-DMF Biochip

Algorithm 2 describes pseudo code of the proposed router. This router is based on computational geometry concepts. We first convert a droplet to unit size droplet and blow up the configuration. Status like busy, free, border are assigned to each cell. Then, a graph is computed from direct paths and if no direct path is possible, edges of convex hull of intersecting obstacles are added. After the graph has all obstacle free edges, we find the shortest path using dijkstra's algorithm.

Algorithm 2: Proposed Router

```
Input  : Source, destination, droplet and configuration
Output: Route and routing time
Find the blowup direction for each obstacle in the configuration;
Blow up the obstacles using blowup direction;
Assign the status to each cell and find overlapped regions;
From source to destination, find direct path and edge to graph $G$;
while edges in graph G are not obstacle-free do
    Identify obstacle in the edge, remove overlapping of this obstacle and assign size
      to each cell;
    Find convex hull for edge start cell, intersecting obstacle and edge end cell and
      edges of computed convex hull to the graph $G$;
Compute edge weights in graph $G$;
Apply Dijkstra's algorithm to find the shortest route from source to destination;
```

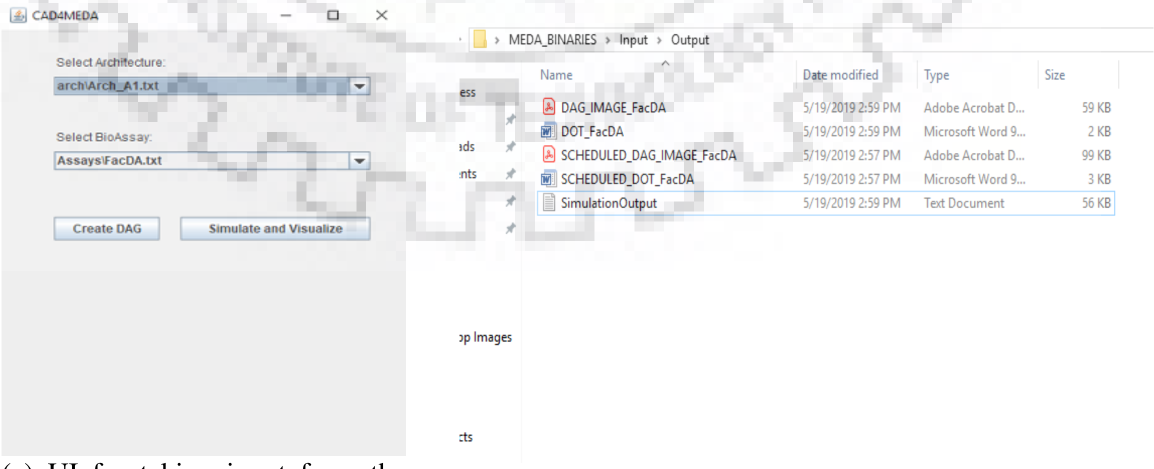
Chapter 7

EMPIRICAL RESULTS

7.1 Demonstration of the Tool Developed Till Now

In this dissertation, we have successfully defined the bioassay file, architecture file and library file. We proposed the new synthesis approach. Out of the three phases: scheduling, placement and routing, we were successful in implementing the first two. Due to a few cases in routing phase, the proposed routing algorithm got complex and out of scope. But the idea of considering routing time with placement has been completed by using constant routing time. We also decided to develop a tool for MEDA-DMF biochip where complete synthesis is done. For now, user gives a bioassay file and an architecture file as input and gets unscheduled, scheduled and placed DAG as output. Screenshots of output can be seen from Figure 7.1 to Figure 7.5.

Figure 7.1 (a) shows the UI of the tool. Here, an architecture file and a bioassay file is selected and with the help of two buttons, unscheduled and scheduled DAG are generated. The output of this tool is saved in a output folder as shown in Figure 7.1 (b).



Select Architecture: arch\Arch_A1.txt

Select BioAssay: Assays\FacDA.txt

MEDA_BINARIES › Input › Output

Name	Date modified	Type	Size
DAG_IMAGE_FacDA	5/19/2019 2:59 PM	Adobe Acrobat D...	59 KB
DOT_FacDA	5/19/2019 2:59 PM	Microsoft Word 9...	2 KB
SCHEDULED_DAG_IMAGE_FacDA	5/19/2019 2:57 PM	Adobe Acrobat D...	99 KB
SCHEDULED_DOT_FacDA	5/19/2019 2:57 PM	Microsoft Word 9...	3 KB
SimulationOutput	5/19/2019 2:59 PM	Text Document	56 KB

(a) UI for taking input from the tool.

(b) Output folder.

Figure 7.1: Input and output for the tool.

Figure 7.2 shows the unscheduled DAG. The information in each node tells node ID and type of node.

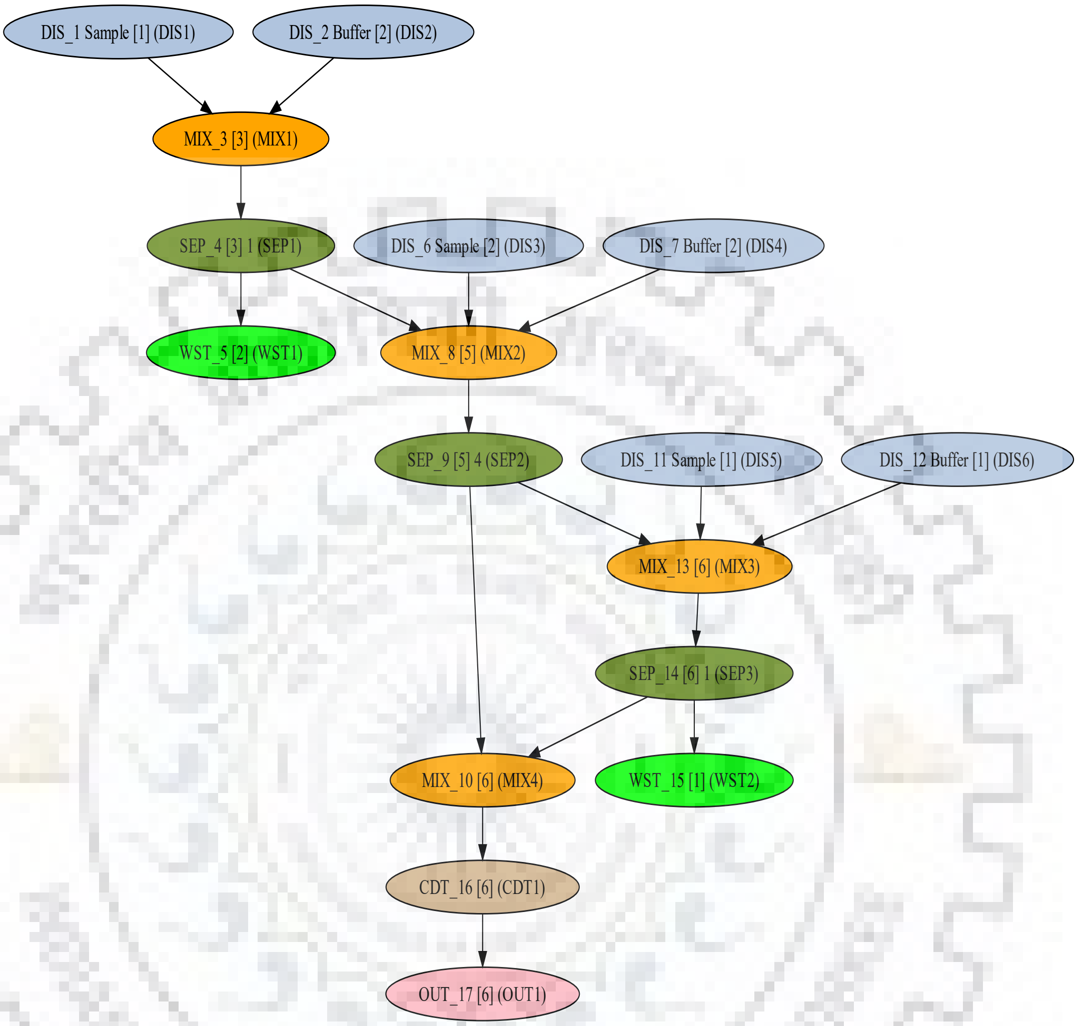


Figure 7.2: DAG file for FacDA.

Figure 7.3 shows the scheduled and placed DAG. There are some labels in scheduled DAG which needs explanation. The first line in each node tells node ID and type of node. The tag *Sched* gives the scheduling time or the exact execution time of an operation. There is a possibility that some droplet reaches a module (mixer) before the other droplet. In this case, *PreStrg* label is used. This reserves the module to store early reaching droplet. Similarly, *Strg* label stores the droplet for some extra time after completion of the operation. The tag *Place* tells the exact location of the module. *TL* stands for top left cell and *BR* stands for bottom right cell of the module. The tag *ResID* tells which non-reconfigurable resource is reserved for the corresponding operation.

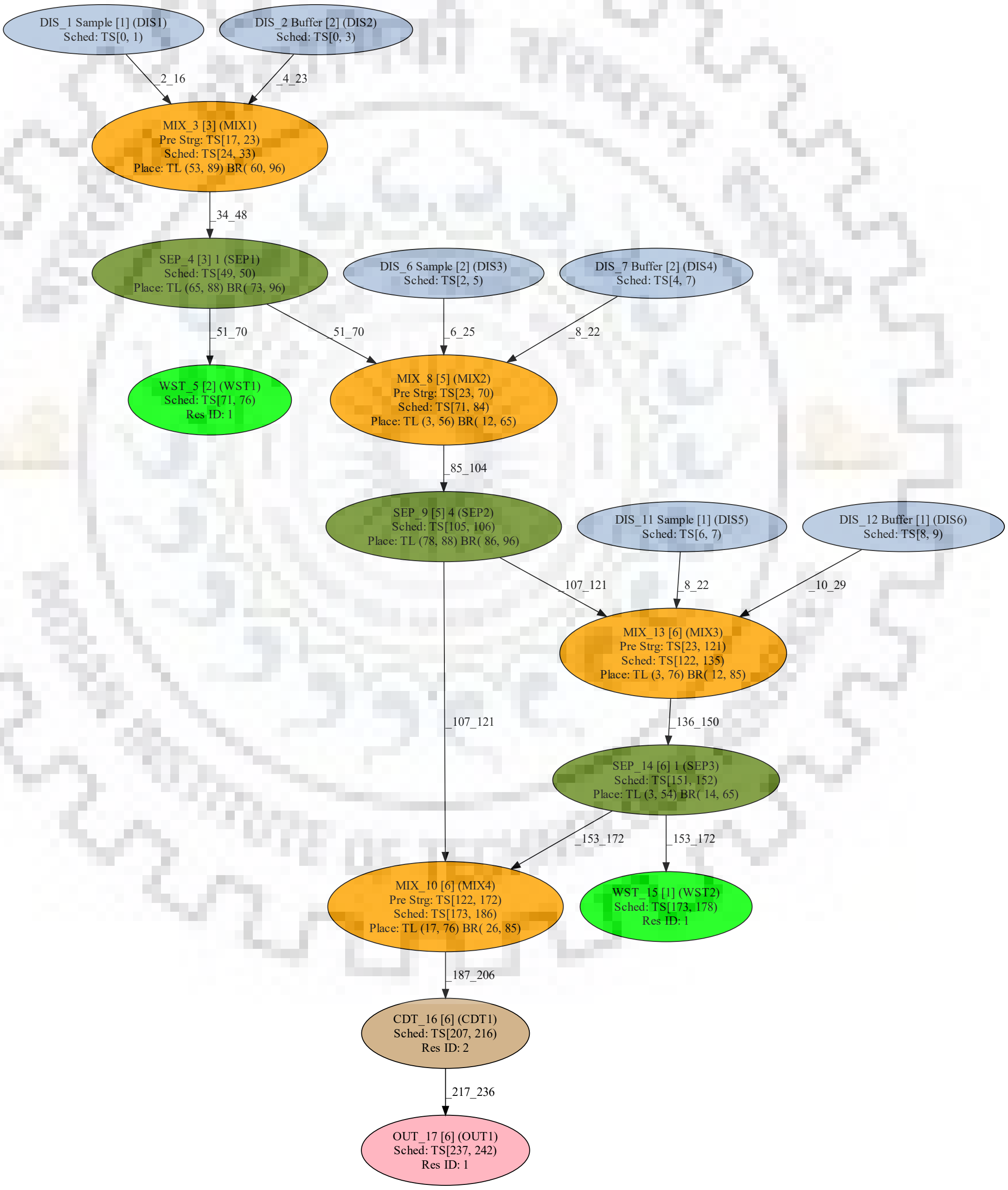


Figure 7.3: Scheduled and placed DAG for FacDA.

Figure 7.4 shows the unscheduled DAG and Figure 7.5 shows the scheduled and placed DAG for Synthetic1.

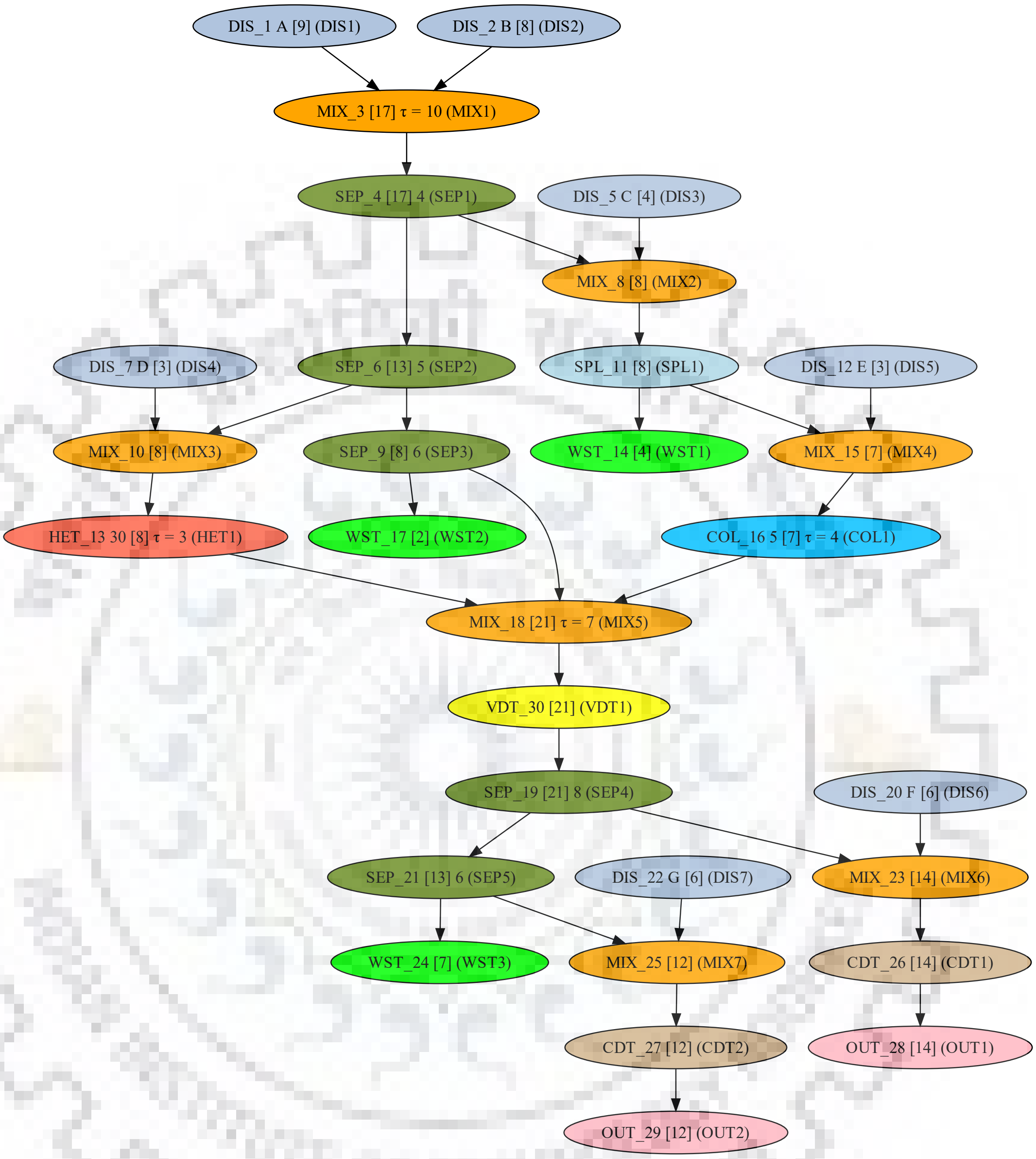


Figure 7.4: DAG file for Synthetic1.

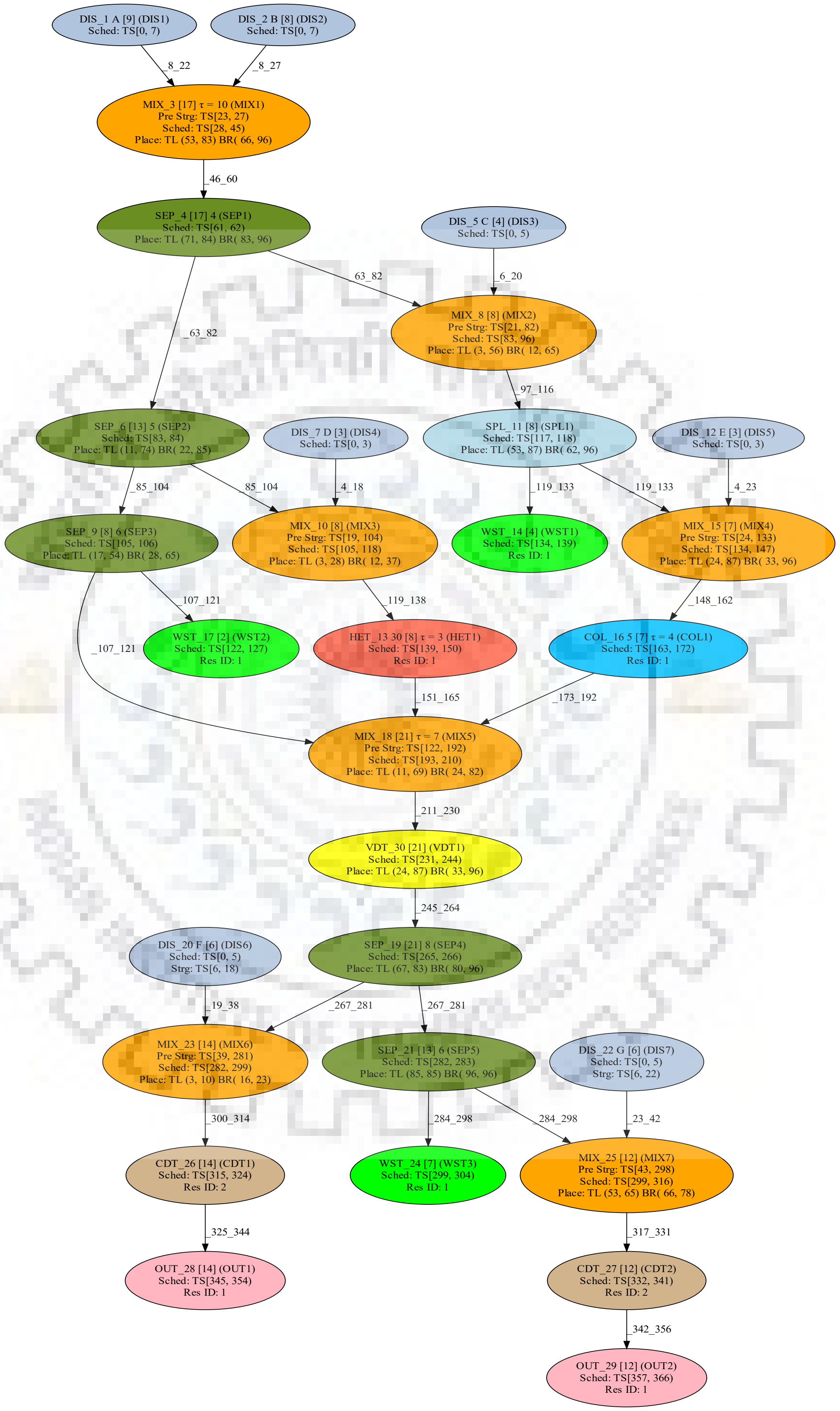
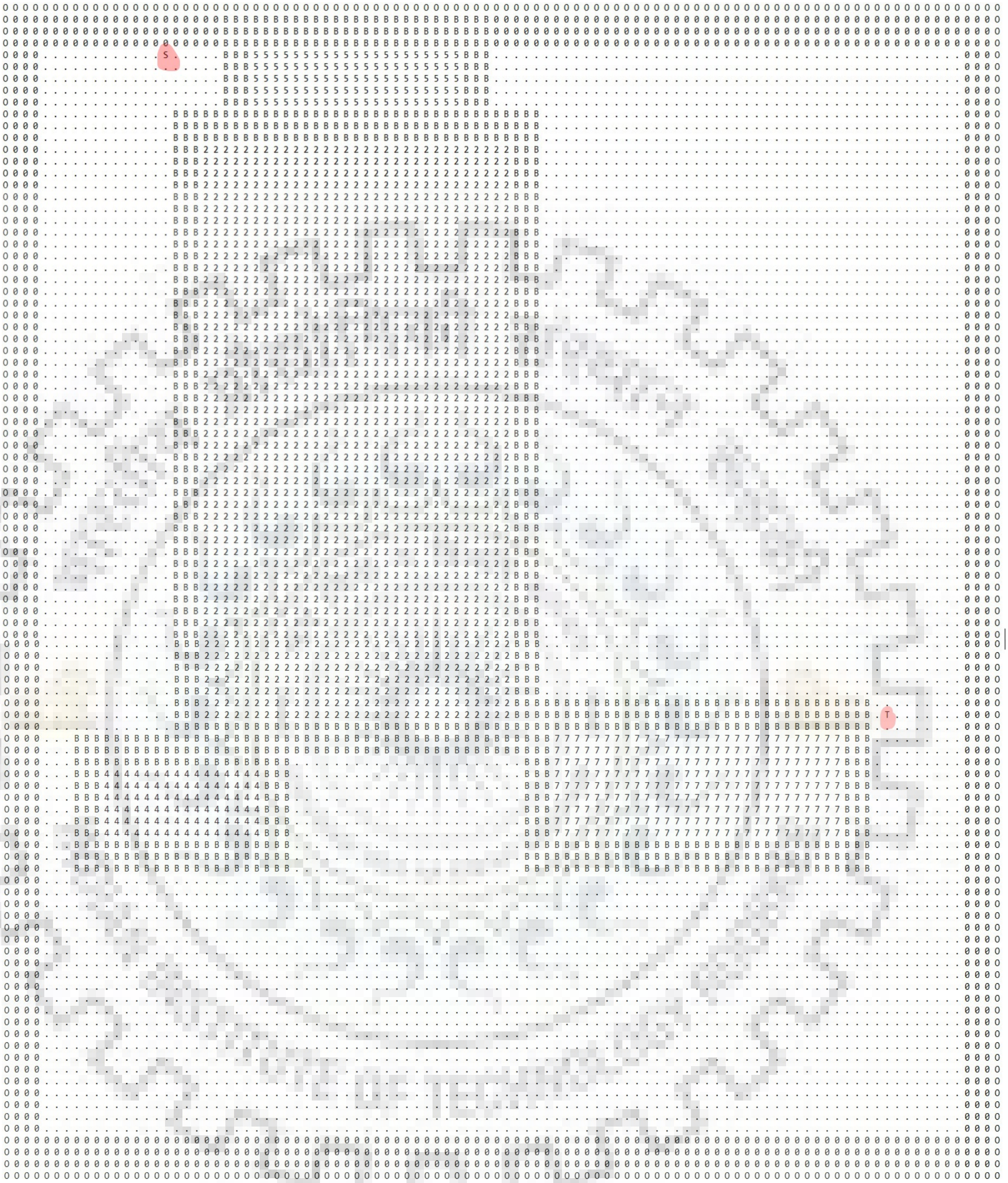


Figure 7.5: Scheduled and placed DAG for Synthetic1.

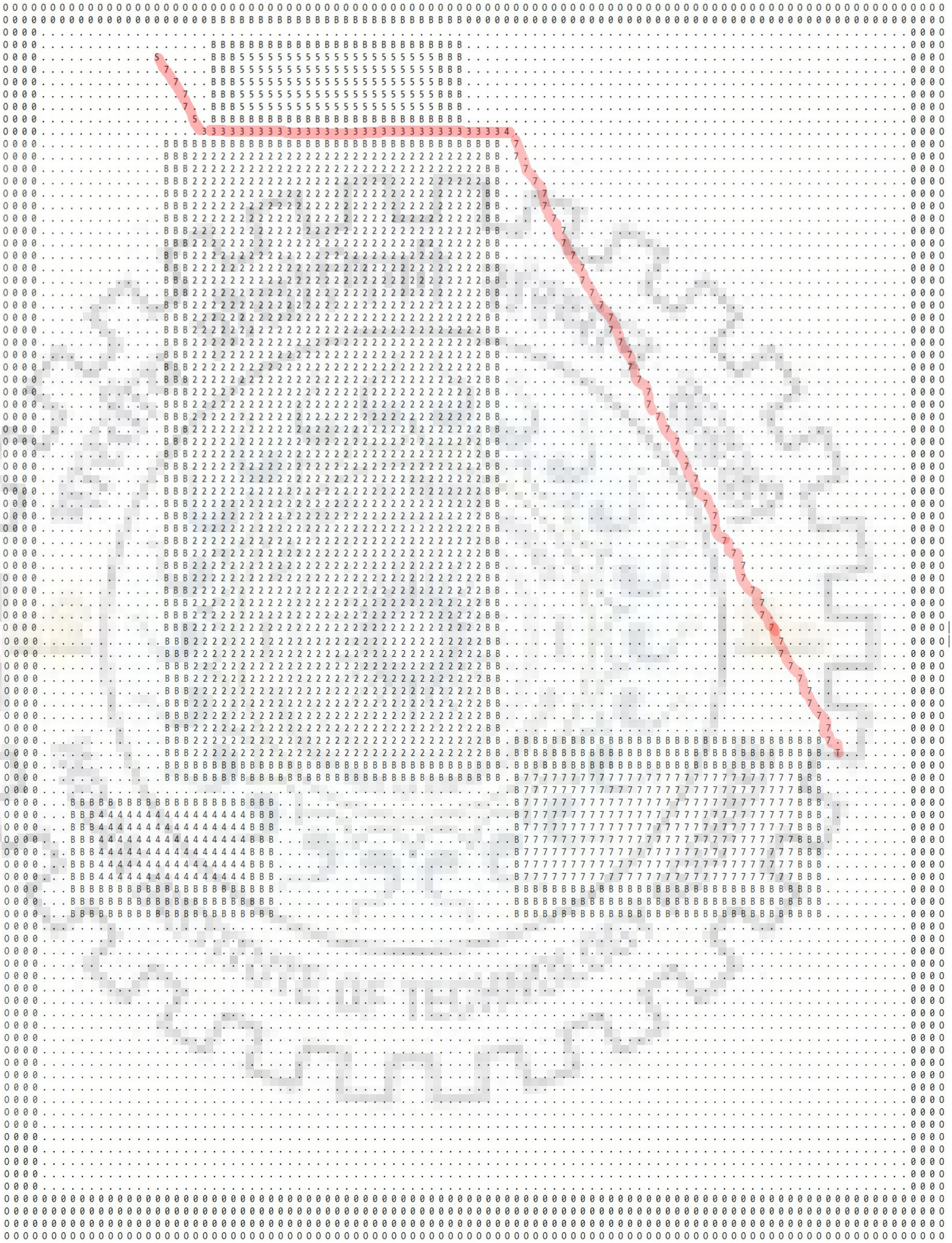
Figure 7.6 shows the configuration of chip before routing. *B* denotes the blownup cells, '.' shows free cells and any numerically denoted cell is obstacle.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B B B B B B B B B B B B B B B B B B B B B B B B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B B B B B B B B B B B B B B B B B B B B B B B B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B B B B B B B B B B B B B B B B B B B B B B B B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 . . . . . . . . . . . . S . . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . T . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . B B B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 7.6: Chip configuration with source *S* and destination *D* before routing .

Figure 7.7 shows the configuration of chip after routing. In Figure 7.7, overlapping of obstacles is removed and path is highlighted. The numerical values on path denotes the size of droplet that can traverse through that channel.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B B B B B B B B B B B B B B B B B B B B B B B B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . S . . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . 7 . . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . 7 . . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . 7 . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . 7 . . B B B 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . 5 . B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B 7 . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 B B . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . T . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 B B B . . . . . . . . . . . . . . . . . . . . . . . . . B 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . . . . . . . . . . . . . . . . . B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 7.7: Chip configuration with computed route.

7.2 Discussions

Here are some problems listed which we faced while implementing the proposed routing algorithm for MEDA-DMF biochip.

- The concept of convex hulls in routing failed when source or target occured inside convex hull of orthogonal obstacle.
- If two obstacles collide for more than once, we were unable to find overlapped regions in less time.
- The case where two obstacles actually don't overlap but lie next to eachother without any space between them was another case which would increase the complexity of proposed algorithm.
- While considering only rectangular modules, when the source lies in blownup region, it gives rise to another case increasing complexity.

The problems discussed above may have solutions but it may increase the complexity of algorithm more than Lee's routing algorithm. This complexity may be reduced by using better data structures or logics.

Chapter 8

CONCLUSIONS AND FUTURE SCOPE

8.1 Conclusions

MEDA-DMF biochips provide a feasible platform for applications such as clinical diagnostics and biomolecular recognition. Synthesis of biochip includes resource binding, scheduling, module placement, droplet routing and actuation sequence generation. Layout and the generated actuation sequence decides the flow of sample droplets into the biochip so as to achieve the desired result such as dilution to a particular concentration or mixing of two or more reagents in a specified ratio.

Scheduling, placement and routing are important steps in the synthesis of biochip. In this dissertation, we have proposed well defined bioassay file, architecture file and library file. We have proposed a new synthesis approach. Due to a few cases in routing phase, the proposed routing algorithm got complex and out of scope. Hence, only scheduling and placement have been successfully implemented. But the idea of considering routing time with placement has been completed by using constant routing time. The work is presented with the help of user interface developed in java. A lot of components, algorithms can be later added to this UI to convert it into a complete tool.

8.2 Future Scope

- One of the important output of any synthesis approach is the activation sequence. In DMF biochip architecture, it is simple to generate electrode actuations but in MEDA-DMF biochip, minimum electrode actuations to traverse the droplets is another study.

- A better priority generator can be implemented which can backtrack the operations and result in better scheduling.

- Finding an appropriate location for placing module in a free rectangle. In this work, module is always placed in bottom left of the free rectangle. Placing of module should be such that the chip breaks into larger free rectangles rather than a number of small free rectangles.

- For operations having more than one parent like mix operation, the time for sources to reach destination should be approximately equal or near to equal. For this, appropriate location is to be found in the free rectangle.

REFERENCES

[1] K. Chakrabarty and F. Su, *Digital Microfluidic Biochips: Synthesis, Testing and Reconfiguration Techniques*. CRC Press, 2007.

[2] W. Thies, J. P. Urbanski, T. Thorsen, and S. Amarasinghe, "Abstraction Layers for Scalable Microfluidic Biocomputing," *Natural Computing*, vol. 7, no. 2, pp. 255–275, 2008.

[3] D. Grissom and P. Brisk, "Path scheduling on Digital Microfluidic Biochips," in *Proc. of the IEEE/ACM Design Automation Conference (DAC)*, 2012, pp. 26–35.

[4] O. Keszocze, Z. Li, A. Grimmer, R. Wille, K. Chakrabarty, and R. Drechsler, "Exact routing for micro-electrode-dot-array digital microfluidic biochips," in *Proc. of the Asia and South Pacific Design Automation Conference*, 2017, pp. 708–713.

[5] Anita, "Geometry-oriented fluid routing algorithm for meda-based digital micro uidic biochips," Master's thesis, Indian Institute of Technology Roorkee, 2017.

[6] Y. Lu, T. Marconi, G. Gaydadjiev, and K. Bertels, "An efficient algorithm for free resources management on the field-programmable gate array," in *Proc. of the Conference on Design, Automation and Test in Europe*, 2008, pp. 1095–1098. [Online]. Available: http://doi.acm.org/10.1145/1403375.1403638

[7] N. A. Sherwani, *Algorithms for VLSI Physcial Design Automation*. Kluwer Academic Publishers, 1998.

[8] Z. Chen, D. H.-Y. Teng, G. C.-J. Wang, and S.-K. Fan, "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. [Online]. Available: https://doi.org/10.1007/s13206-011-5408-5

[9] G.-R. Lu, B. B. Bhattacharya, T.-Y. Ho, and H.-M. Chen, "Multi-level droplet routing in active-matrix based digital-microfluidic biochips," in *Proc. of the 23rd Asia and South Pacific Design Automation Conference*, 2018, pp. 46–51.

[10] "Microfluidics @ ucr," Online, http://microfluidics.cs.ucr.edu/index.html.

[11] Z. Li, K. Y. Lai, J. McCrone, P. H. Yu, K. Chakrabarty, M. Pajic, T. Y. Ho, and C. Y. Lee, "Efficient and Adaptive Error Recovery in a Micro-Electrode-Dot-Array Digital Microfluidic Biochip," *IEEE TCAD*, vol. 37, no. 3, pp. 601–614, 2018.

[12] Z. Li, K. Y. Lai, P. Yu, K. Chakrabarty, T. Y. Ho, and C. Y. Lee, "Structural and Functional Test Methods for Micro-Electrode-Dot-Array Digital Microfluidic Biochips," *IEEE TCAD*, vol. 37, no. 5, pp. 968–981, 2018.

[13] Z. Zhong, Z. Li, and K. Chakrabarty, “Adaptive and Roll-Forward Error Recovery in MEDA Biochips Based on Droplet-Aliquot Operations and Predictive Analysis,” *IEEE TMSCS*, vol. 4, no. 4, pp. 577–592, 2018.

[14] M. Shayan, S. Bhattacharjee, T. C. Liang, J. Tang, K. Chakrabarty, and R. Karri, “Shadow Attacks on MEDA Biochips,” in *Proc. of the ICCAD*, 2018, pp. 73–80.

[15] V. Shukla, F. A. Hussin, N. H. Hamid, N. B. Z. Ali, and K. Chakrabarty, “Offline Error Detection in MEDA-Based Digital Microfluidic Biochips Using Oscillation-Based Testing Methodology,” *Springer JETTA*, vol. 33, no. 5, pp. 621–635, 2017.

[16] T.-C. Liang, M. Shayan, K. Chakrabarty, and R. Karri, “Execution of Provably Secure Assays on MEDA Biochips to Thwart Attacks,” in *Proc. of the ASP-DAC*, 2019, pp. 51–57.

[17] Z. Li, K. Y.-T. Lai, K. Chakrabarty, T.-Y. Ho, and C.-Y. Lee, “Droplet Size-Aware and Error-Correcting Sample Preparation Using Micro-Electrode-Dot-Array Digital Microfluidic Biochips,” *IEEE TBCAS*, vol. 11, no. 6, pp. 1380–1391, 2017.

[18] S. Saha, D. Kundu, S. Roy, S. Bhattacharjee, K. Chakrabarty, P. P. Chakrabarti, and B. B. Bhattacharya, “Factorization Based Dilution of Biochemical Fluids with Micro-Electrode-Dot-Array Biochips,” in *Proc. of the ASP-DAC*, 2019, pp. 462–467.

[19] T. C. Liang, Y. S. Chan, T. Y. Ho, K. Chakrabarty, and C. Y. Lee, “Sample Preparation for Multiple-Reactant Bioassays on Micro-Electrode-Dot-Array Biochips,” in *Proc. of the ASP-DAC*, 2019, pp. 468–473.

[20] W. Chung, P. Cheng, Z. Li, and T. Ho, “Module Placement under Completion-Time Uncertainty in Micro-Electrode-Dot-Array Digital Microfluidic Biochips,” *IEEE TMSCS*, vol. 4, no. 4, pp. 811–821, 2018.

[21] Z. Chen, D. H. Y. Teng, G. C. J. Wang, and S. K. Fan, “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.

[22] M. Alistar and P. Pop, “Towards Droplet Size-Aware Biochemical Application Compilation for AM-EWOD Biochips,” in *Proc. of the MEMS/MOEMS (DTIP)*, 2015, pp. 1–6.

[23] O. Keszöcze, Z. Li, A. Grimmer, R. Wille, K. Chakrabarty, and R. Drechsler, “Exact Routing for Micro-Electrode-Dot-Array Digital Microfluidic Biochips,” in *Proc. of the ASP-DAC*, 2017, pp. 708–713.

[24] G.-R. Lu, C.-H. Kuo, K.-C. Chiang, A. Banerjee, B. B. Bhattacharya, T.-Y. Ho, and H.-M. Chen, “Flexible Droplet Routing in Active-Matrix Based Digital Microfluidic Biochips,” *ACM TODAES*, vol. 23, no. 3, pp. 37:1–37:25, 2018.

[25] Z. Li, K. Y.-T. Lai, P.-H. Yu, K. Chakrabarty, T.-Y. Ho, and C.-Y. Lee, “Droplet Size-Aware High-Level Synthesis for Micro-Electrode-Dot-Array Digital Microfluidic Biochips,” *IEEE TBCAS*, vol. 11, no. 3, pp. 612–626, 2017.

[26] G. Wang, D. Teng, and S. Fan, “Digital microfluidic operations on micro-electrode dot array architecture,” *Nanobiotechnology*, vol. 5, no. 4, pp. 152–160, 2011.

[27] N. Masoudi, “Geometric path-planning algorithm in cluttered 2D environments using convex hulls,” Master’s thesis, Clemson University, 2017.

[28] T.-J. Pan and R. C. Luo, “Motion planning for mobile robots in a dynamic environment with moving obstacles,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 1990, pp. 578–583.

[29] M. J. Jebrail, V. N. Luk, S. C. Shih, R. Fobel, A. H. Ng, H. Yang, S. L. Freire, and A. R. Wheeler, “Digital microfluidics for automated proteomic processing,” *JoVE (Journal of Visualized Experiments)*, no. 33, p. e1603, 2009.