Benchmarking Microfluidic Design Automation Flows

Ashton Snelgrove, Skylar Stockham, Pierre-Emmanuel Gaillardon
Electrical and Computer Engineering
University of Utah
Salt Lake City, Utah, USA
pierre-emmanuel.gaillardon@utah.edu

***Abstract*—In this paper, we propose a methodology for measuring figures of merit relevant to microfluidics practitioners. We also present a benchmark suite for microfluidics design automation (MFDA). The suite is composed of generated circuits designed to challenge tools for specific figures of merit identified in the measurement methodology. We survey the MFDA literature to evaluate the state-of-the-art for benchmark and measurement methodologies. We include in the benchmark distribution a complete set of transcriptions of the major benchmarks used in the MFDA literature, including designs previously unavailable. Benchmarks are distributed in the major hardware description languages along with reported measurements**

***Index Terms*—Microfluidics, Electronic Design Automation, 3D Printing, Placement and Routing**

I. Introduction

Microfluidics concerns the precise manipulation of small amounts of fluids at the nanoliter scale. This is used to create so-called "lab-on-chips" which consolidate chemical or biological assays into small automated devices. The goal of microfluidic design automation (MFDA) is to introduce computer tools and algorithms to automate aspects of the design process. [1]

The MFDA community is faced with several related challenges. The community is lacking an available set of comprehensive, shared, and open-source benchmarks. Previous works identified this problem and began a collection of open-source benchmarks[2][3] However, this still remains a limited set that does not bring together all of the major benchmarks currently in use in the literature, and has not seen adoption. Current benchmarks do not reflect the scale currently achieved in electronic design. A set of benchmarks is needed that will challenge MFDA at scale.

The community is also lacking a clear consensus for appropriate figures of merit, in particular metrics aligned with the needs of microfluidics practitioners[4]. A pressing concern is the adoption of new manufacturing techniques such as 3D printing which challenge existing assumptions and requirements.

To address these issues, this work proposes:

This material is based upon work supported by the National Science Foundation under Grant No. 2245494. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1) A complete set of transcriptions of the major benchmarks in use in the MFDA literature.
2) A methodology for measuring figures of merit relevant to microfluidics designers.
3) A set of synthetic benchmarks designed to challenge tools on specific figures of merit.
4) Initial reported measurements for each benchmarks.

Our proposed figures of merit fall into three categories: practitioner needs, manufacturability, and optimization. Practitioners are particularly interested in tool runtime and meeting design constraints[4]. Manufacturing concerns include area/volume and component density. Optimization concerns are related to the output of synthesis algorithms, including control line minimization and component counts. Our methodology also seeks to address the question of comparing traditional planar manufacturing and 3D printing.

Our proposed benchmark suite provides a range of synthetic benchmarks designed to challenge specific metrics in the placement and routing (PNR) stages of design automation.

The remainder of our paper is organized as follows. Section II evaluates previous measurement methodologies in the literature. Section III proposes a methodology for measuring and evaluating MFDA tools. Section IV surveys the common benchmarks used in the literature. Section V proposes a set of synthetic benchmarks. Section VI concludes the paper.

II. Current use of metrics and figures of merit

A. Survey of commonly reported metrics in the literature

We surveyed 39 papers which were referenced from two literature reviews, and evaluated each for metrics reported and benchmarks used. Papers were taken from Table 1 in Sanka et al. [3] and from Tables 1 and 2 in Huang et al. [1]. We noted which metrics and which benchmarks were used in each paper. All papers were targeting planar devices implemented in PDMS.

Table I summarizes the results for metrics used. The most commonly used metric was runtime of algorithm or tool, with 87% reporting. 79% reported which type of compute resources were used: usually the processor model, available memory, and operating system. Programming language used for implementation was usually given. Tools using an external solver usually specified which was used (such as an integer

Metric	# papers	% papers
Component count	26	67%
Number of valve switching states	7	18%
Valve or control line count	21	54%
Number of flow/control intersections	9	23%
Total linear channel length	20	51%
Delay or latency	10	26%
Area	16	41%
Algorithm specific synthesis metrics	12	31%
Provides design rules or dimensions	11	28%
Runtime (seconds)	34	87%
Specifies compute resources used	31	79%
Success/failure rate of algorithm	4	10%

TABLE I
SURVEY OF METRICS USED IN 39 PAPERS REFERENCED FROM [1][3]

linear programming solver). Runtime is easily quantifiable, and non-specific to any particulars of algorithm or benchmark. However, while most papers reported descriptions of the environment, comparing run time between tools is difficult due to no clear scaling between different computing resources. Huang et al.[1] assigned a general "fast, medium, slow" rating to each algorithm in their review.

The next most commonly specified metric was some type of operational component count, with 67% reporting. 50% of papers which specified a component count also specified a count of each types of components (such as mixers, heaters, etc.). This metric is difficult to compare due to potentially conflicting meanings. Some papers are simply specified the size of the benchmark, such as those optimizing placement and routing without component minimization. Other papers were specifically attempting to do resource allocation and scheduling, and minimizing the components used in the schedule.

54% of papers gave a count of valves or control lines. Intersections between flow and control layers was reported in 23%. Both of these represent synthesis targets for the flow/control layer interaction, for control line minimization and planarity layout optimization respectively. Control input ports in particular represent a resource bottle neck.

Size information was present in many of the papers. 51% of papers specified a total linear channel length. 26% specified a delay or latency metric of a synthesized schedule. 41% specified an area measurement of the final device. However, only 28% specified details of design rules, component or channel dimensions, or device specifics. This is a potential blind spot in comparing dimensions metrics with possible unknown dimensions.

Finally, 31% gave other metrics specific to the algorithm. Papers sometimes provided their own reference algorithm implementation to compare against, stating uniqueness of their target optimization.

B. The move to 3D printing

All of the papers surveyed represent design flows targeting planar PDMS manufacturing. Metrics such as minimizing control/flow layer intersection for planarity are imposed by the underlying manufacturing technology. New manufacturing techniques stemming from advances in 3D printing challenge existing limitations, but also introducing new constraints[5]. Volume in planar devices is defined by the two-dimensional area with a fixed depth of the feature. Three-dimensional features can utilize varying depths to reduce feature surface area and footprint. Multi-layer routing, typical of electronic manufacturing, removes the routing restrictions imposed by a single pair of control and flow layers.

Some challenges remain the same between technologies. Input/output line utilization remains a problem, perhaps even worse in 3D printed devices due to the reduced surface area of a three-dimensional device and the fixed width and height of the print area in a stereolithography (SLA) printer. Innovations in connection miniaturization are promising for increasing the density of external connections [6]. Control line and valve count minimization also remains a concern. Recent work proposes transistor like microfluidic structures, bringing the potential for on-chip control logic[7], but some amount of external control will continue to be required. Minimizing total channel length will remain a consistent concern.

C. Metrics of relevance to designers

McDaniel et al[4] interviewed approximately 100 lab-on-chip designers to determine what factors are relevant to their work. Of primary concern was correct device function, meeting design constraints, reliability, legibility, and testability. They are critically interested in maintaining control of the design process. Designers recognized a need for fast algorithms and supplemental tools which boost productivity and address specific pain points of their workflows. Slow algorithms would not be adopted, and designers showed no trust or interest in full turn-key solutions. Metrics associated with EDA such as channel length and skew, were considered of little utility. Transport delay has little bearing when assay times are measured in minutes or hours. Minimizing area is not a significant cost concern[4]. Reducing reagent and sample volumes and waste is often cited as a benefit of lab-on-chips[1], which places interest on minimization of channel lengths. However, channel length is not considered relevant by device designers[4].

The adoption of 3D printing introduces new demands on tools. Given the desire for fast algorithms, it is likely faster prototyping turnaround times would also be desirable. For SLA 3D printing, print time is determined by device depth[5], which could be a target metric for minimization. SLA printing works within a fixed print area, which places a manufacturing constraint on area that may not be previously present when working with planar technologies. SLA printed devices require post-processing flushing of uncured resin trapped in the device during printing. Channel lengths are a major contributor to flushing time.

As is the case with EDA designers, area or timing optimizations are not a problem right up until constraints are violated. MFDA tools currently do not have a format for design constraint specification, which will be required for future work. Synthesis algorithms require a level of trust that users

do not appear to be willing to give[4]. Future development of tools for verification may improve this situation.

III. Proposed measurement methodology

We identify three groups of metrics: universal metrics that affect user workflow, metrics for synthesis and optimization, and metrics for physical design.

Optimization is only of practical use when it is in service of meeting design constraints. When comparing algorithms we often target absolute value improvement on a metric. Absolute improvement reflects on the capability of a tool, but does not necessarily translate to improvements in outcomes to meet design constraints. Measurements are also often taken without first doing predictions of outcome. We propose that benchmarks also include a set of design constraints of varying levels of strictness that challenge tools in multiple conditions.

Testing and quality assurance metrics such as validation time are also relevant metrics, but are not within the scope of this work. Finally, the implicit metric for any tool is a need for maintaining correctness. This needs to be explicitly noted here, as design verification is not currently an explored problem in MFDA. Verification is beyond the scope of this work, but must be addressed in the future.

A. Universal metrics

Metrics applicable to all scenarios. These metrics are critical to two identified user priorities, fast runtime and meeting design constraints [4].

- Compute environment. Results should be placed in the context. Papers should report CPU model and speed, total system memory, programming language, and operating system.
- Runtime; in seconds. Despite the difficulty in comparing time between compute environments, runtime remains a critical metric. Ideally, papers should report runtimes for both proposed tools and the tools being compared against, all run in the same environment. Runtimes should be reported for each flow step, so that comparisons can be made for individual steps if some steps were not performed.
- Failure to generate a solution, meet user design constraints, or pass manufacturer's technology design rules; binary success/failure. If an algorithm fails to complete or cannot provide a solution, this should be noted. If algorithms are non-deterministic, percent failure rate should be given.

B. Synthesis metrics

Metrics applicable to tools doing high level device synthesis. Expected tool input is a high level description of device operation. Expected tool output is a netlist. Goals for synthesis are optimization.

- Component count. Synthesis algorithms performing allocation and mapping for schedule. Specifying the types of allocated component is useful.
- Valve count. Synthesis algorithms performing control line minimization.
- Input/output port count. Fluid transport requirements and control line minimization.

C. Physical design metrics

Metrics applicable to physical device generation. Expected tool input is a netlist. Expected tool output is manufacturing plans. Goals for physical design is manufacturability. Benchmarks are defined with the following set of characteristics:

- Component, valve, and IO counts. Reported values used to describe scale of benchmark problem.
- Minimal channel feature size; in micrometers. Establishing bounds on the manufacturing technology to adjust for differences in sizes.

The figures of merit themselves are reflective of the physical characteristics of the manufactured device.

- Simple dimensions height, width, and depth; in either micrometers or multiple of minimum feature size. For SLA printing, depth is the primary factor impacting print time.
- Spacial utilization; percent of device size. For two-dimensional devices, this a percentage of area occupied by components and channels. For three-dimensional devices, this is percentage of volume.
- Effective planar area; in either micrometers or multiples of feature size. For two-dimensional devices, this is the simple area, height $\times$ width. For three-dimensional devices, multiply the simple area by the number of routing layers to get an approximate equivalent planar area. If routing is not done in a layered fashion, replace number of layers with depth divided by the minimum feature size.
- Total and maximum channel length; in either micrometers or multiple of feature size. For SLA printing, this metric is also related to post-print flush time.

IV. Current state of the art for benchmarks

A. Commonly used benchmarks

Table II lists the frequency of benchmarks used in the surveyed papers. No benchmark was used in a majority of MFDA papers surveyed. Three sets of benchmarks appear most often in the papers surveyed. The first is a set originating from Duke University [8]. It consists of three designs derived from real-life assays: polymerase chain reaction (PCR), in-vitro diagnostic (IVD), and colorimetric protein assay (CPA).

The second is a set originating from Technical University of Denmark (DTU)[9]. It consists of five synthetic designs of increasing size from 10 to 50 components.

A third set consists of devices with similar structure derived from published fluidic devices[10]–[14], first appearing in papers from Technical University of Munich (TUM)[15].

Of these benchmarks the PCR benchmark is referenced in 44% of papers surveyed. 26% of papers used benchmarks from two sets, 38% used benchmarks from only one set, and 36% used no benchmarks from any set. 56% of papers used a benchmark unique to that paper.

Benchmark	# papers	% papers
PCR[8]	17	44%
IVD[8]	14	36%
CPA[8]	12	31%
DTU Synthetic 1-5 [9]	12	31%
Kinase[11]	4	10%
MNAcid Process [12]	4	10%
MRNA[10]	6	15%
ChIP[13]	6	15%
HIV1[14]	3	8%
Other paper specific benchmark	22	56%

TABLE II
SURVEY OF BENCHMARKS USED IN 39 PAPERS REFERENCED BY [3][1]

Benchmarks specific to papers were typically random graphs or synthetic benchmarks designed for the paper. No reproducible details of the structure of these benchmarks was given.

B. Benchmark analysis

These benchmarks suffer from several drawbacks. The Duke and DTU sets are given as abstract sequence graphs of fluidic operations. This is an appropriate input for a high level synthesis algorithm, such as allocating the sequence onto shared resources and doing scheduling. However, no standard synthesized netlist is available for these designs to use in evaluating stages such as placement and routing. The distributed format of these benchmarks is an image showing the sequence graph. No hardware description language (HDL) is known to the authors which has been used to encode the graph - the image is the canonical definition. Both of these sets are frequently cited at the original URLs - both of which are no longer directly available, nor are they publicly archived. Recent papers increasingly cite older papers that use the benchmarks rather than the original URLs.

The TUM set is limited by its focus on benchmarks that represent a single class of design. The benchmarks are all of designs utilizing a circular reaction/mixer structure. The benchmarks themselves are taken from peer reviewed papers with clear device images and operational descriptions which can be transcribed. The example netlists given on the Cloud Columba website [15] utilizes a custom HDL, which specifies two possible components, the circular reactor and a reservoir. However, the examples are not an exact match to the devices presented in the source papers. Simplifications were made to the device layout, particularly in combining control and pump lines. Some simplifications appear to change the function of the design from the procedure described in the original paper.

A set of benchmarks *Parchmint* has been proposed as a common standard[2]. Parchmint provides netlists in a consistent JSON format. JSON does have an advantage of being easily read by libraries in any programming language. The benchmarks can also be found in the MINT HDL netlist format. The suite consist of benchmarks aggregated from those used in the authors' previous works. This consist largely of benchmarks originally designed for the Fluigi design flow [16]. Parchmint also proposes a synthesized netlist of some of the DTU synthetic benchmarks. However it is unclear which of the seven designs presented in Parchmint is associated with which of the five designs in the original DTU and what modifications where made. Several additional benchmarks were transcribed from device images presented in assay design papers, including one[14] also used in the TUM set. The transcriptions do not appear to reflect the exact structure presented in the original paper, but have simplifications or modifications. At time of writing, only one paper was found to cite and use the Parchmint benchmarks, also written by the primary author of the benchmark suite[17].

C. Necessary characteristics

The literature survey identified several important characteristics:

1) Applicability. The Duke and DTU sets are operation sequence graphs, appropriate as input to a high level synthesis flow. The TUM and Parchmint sets are netlists, appropriate as input to placement/routing and manufacturing flows. The two types are not targeting the same goals.
2) Availability. The Duke and DTU sets were referenced by online location and became unavailable when the original website was no longer hosted. A copy of the Duke set has since been shared online[8], and the DTU sets were reprinted in a dissertation[9]. Benchmarks transcribed from fluidic devices presented in published papers could be retranscribed as long as the papers remain accessible.
3) Reproducibility. A majority (56%) of surveyed papers used benchmarks unique to the work with no way to reproduce the benchmark.
4) Consistency. Transcribed benchmarks from the TUM and Parchmint sets did not match the exact structure presented in the original device image.
5) Accuracy. Benchmarks need to capture design constraints. Transcriptions should accurately match dimensions in the original work. The MINT and Columba HDLs as well as the Parchmint JSON structure include dimensional information for channels and components, necessary for accurate representation. None of these formats are capable of capturing the temporal information inherent to the operation sequence graphs from DTU and Duke. Because valve control logic largely lies external to the bounds of the chip, this information must also be captured.
6) Interoperability. Research groups developed benchmarks that would be consistently used within their own work, but rarely used in work by other groups. Project specific HDLs inhibit access without tools to convert between formats.
7) Utility. Benchmarks should demonstrate the key figures of merit. While inspired by real assays, the Duke and DTU benchmarks are only partially representative of actual assays. The TUM benchmarks do not provide a sufficient variety to demonstrate a range of capabilities. The use of a limited set of benchmarks could lead

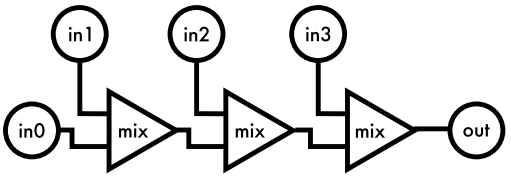

Fig. 1. Example of an n-stage chain of mixers. Benchmark challenges linearly scaling the number of components and ports.

to overfitting algorithms to the benchmarks rather than actual use cases.

8) Complexity. While the PCR benchmark is the most frequently used, it consists of only 7 cascaded mixing operations. This is valuable as a baseline, but does not demonstrate the complexity of design that MFDA tools are promising to handle[1].

V. Proposed benchmark methodology

We now propose our physical design benchmark suite. The suite consists ten benchmarks designed to challenge figures of merit in the automation of physical design. A summary of the new proposed benchmarks and their features is presented in Table III

The repository also includes a transcription of the Duke and DTU sequence graphs. Transcriptions from the original designs are included for benchmarks discussed in section IV and the Parchmint benchmarks. Designs are rendered when possible in the Columba HDL[15], MINT HDL, Parchmint[2], and Verilog.

The benchmarks are distributed as a set of netlists through a GitHub repository.

A. Functional descriptions

The *complete* and *complete-bipartite* benchmarks are complete graphs - all vertices are connected with an edge to every other vertices. Each vertex is a port, and each edge connects a pass through connection directly between ports.

The *chain* benchmark is a sequence of small chambers connected in series. One input port sources the first element in the chain, and one port sinks the last element.

The *mixer-chain* is a sequence of two-to-one mixers connected in series (Figure 1). One input to the mixer comes from the previous stage, and the other input comes from an input port. One input port sources the first element in the chain, and one port sinks the last element.

The *binary-tree* is a full balanced binary tree of depth n. Each node is a two-to-one mixer. There is one output port, and 2^n input ports. Each input port is connected to one input of a mixer.

The *gradient-generator* implements an i input o output gradient generator (Figure 2. Each input port is connected to a serpentine channel. Serpentine channels are place in layers. Each layer has one more layer serpentine than the previous layer. Each serpentine in a layers is connected by a shared channel. Each output is connected to an serpentine connected to the final shared channel.

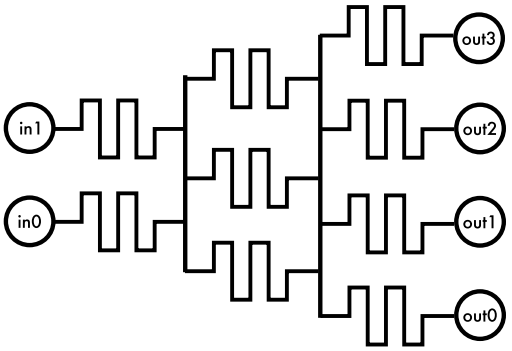

Fig. 2. Example of a 3-stage 2-input gradient generator. Benchmark designed to challenge geometric growth of area utilization.

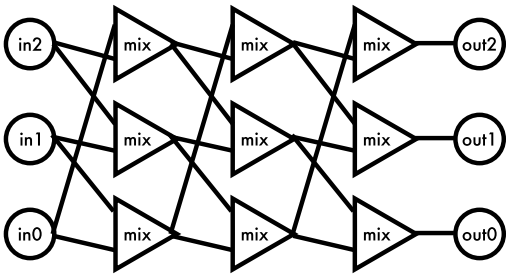

Fig. 3. Example of a 3-wide 3-long braid. Benchmark is designed to challenge linear scaling with mild congestion.

The *braid* implements repeated layers of gates with interwoven fanout (Figure 3). Each layer consists of w nodes with fan-in 2 and fan-out 2. Each node i, fans out into the next layer nodes i and $i+1 \mod w$. The braid is fed by w source ports and w sink ports.

The *multiplexer* implements a full balanced 2^n-to-one multiplexer tree. A core two-to-one multiplexer is made of two valves fed from two input ports, connected to a shared output channel. The larger multiplexer is constructed with a tree of core multiplexers. Each layer of the tree has two control lines shared among all multiplexers in the layer.

The *crossbar* consists of two sets of channels fed from ports, a vertical and horizontal set. Each horizontal channel is connected through a valve to each vertical channel. Connections can be made between a vertical port and a horizontal port by activating the intersecting valve.

B. Design methodology

The largest benchmarks currently in use are small compared to contemporary expectations for EDA. We identified a need to test tools at different design scales and for different design constraints. Benchmarks were selected with different scaling patterns.

Structures were chosen to challenge specific patterns and cases for metrics. The *complete* and *complete-bipartite* graphs were chosen to challenge the densest possible routing. A mix of planar and non-planar graphs were selected to challenge different levels of edges crossing. Trees are commonly found

Name	Vertices	Edges	IO	Challenge
Complete bipartite $K_{n,m}$	0	$2mn$	$n+m$	dense routing, high fanout degrees, high intersections, symmetric
Complete graph K_n	0	n^2	n	dense routing, high fanout degrees, high intersections, symmetric
Chain n	n	$n+2$	2	long planar paths, low IO
n-stage mixer chain	n	$2n$	$n+2$	long planar paths, linear component count, high IO/component connection.
Binary tree n	2^n-1	2^n-1	2^n+1	symmetric planar connections, exponential component count
Gradient Generator i, o	$\sum_{k=i}^{o} k$	$o-i$	$i+o$	symmetric planar connections, exponential component count
Braid w, l	wl	$2wl$	$2w$	linear component count, non-planarity, low intersection
Multiplexer n	2^n-1	2^n-1	2^n+2n+1	planar, linear scale control line, exponential scale component count
Crossbar m, n	mn	$4mn$	$m+n+mn$	non-planar, geometric scale control line and component count, symmetric
Nucleic Acid Processor [12] n	$16n+6$	$14n+25$	$24+n$	typical circular reactor circuit, n parallel reactors

TABLE III
PROPOSED SYNTHETIC BENCHMARKS FOR PHYSICAL DESIGN.

structures, with the *multiplexer* and *gradient generator* representative of practical usage.

Benchmarks are intended to be reproducible from the description in this paper. Common recognizable graph types were used where applicable. Random graphs were avoided - beyond the availability and reproducibility problems, designs are generated in structured ways by human designers, not randomly. The proposed benchmarks do suffer from artificial symmetry and regularity due to the synthetic origin. Additional assay transcriptions are needed, but beyond the scope of this work.

The nucleic acid processor benchmark[12] is representative of the class of design. As specified in the original source, the design already has all control lines shared between parallel stages, with no simplification needed. Scaling the design by increasing parallel stages will not change the structure presented in the source.

VI. Conclusion

MFDA is still a growing discipline, and our survey of the literature shows the use of benchmarks and figures of merit to be fragmented and inconsistent. Previous attempts to create a shared benchmark suite or consistent figures of merit have so far been unsuccessful.

In this work we have proposed a methodology for measuring figures of merit for MFDA tools, with particular attention to the needs of microfluidics designers and emerging manufacturing technologies. Our proposed benchmark suite aims to address the issue of benchmarking design scaling in physical design. We have also included transcriptions of the major benchmarks used in the literature. Future work is needed, particularly in the area of accurate transcription of a broad range of published assays.

We hope that our efforts here improve the accessibility of the current benchmarks in use to the community. Benchmarks in the major HDLs and comparative results for major MFDA tools are distributed through GitHub at https://github.com/utah-MFDA/mfda_benchmarks

References

[1] X. Huang, T.-Y. Ho, *et al.*, "Computer-aided design techniques for flow-based microfluidic lab-on-a-chip systems," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–29, 2022. DOI: 10.1145/3450504.

[2] B. Crites, R. Sanka, *et al.*, "Parchmint: A microfluidics benchmark suite," in *2018 IEEE International Symposium on Workload Characterization (IISWC)*, 2018, pp. 78–79.

[3] R. Sanka, B. Crites, *et al.*, "Specification, integration, and benchmarking of continuous flow microfluidic devices: Invited paper," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–8. DOI: 10.1109/ICCAD45719.2019.8942171.

[4] J. McDaniel, W. H. Grover, and P. Brisk, "The case for semi-automated design of microfluidic very large scale integration (mVLSI) chips," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 2017, pp. 1793–1798. DOI: 10.23919/DATE.2017.7927283.

[5] L. A. Pradela Filho, T. R. L. C. Paixão, *et al.*, "Leveraging the third dimension in microfluidic devices using 3d printing: No longer just scratching the surface," *Analytical and Bioanalytical Chemistry*, 2023. DOI: 10.1007/s00216-023-04862-w.

[6] H. Gong, A. T. Woolley, and G. P. Nordin, "3d printed high density, reversible, chip-to-chip microfluidic interconnects," *Lab on a Chip*, vol. 18, no. 4, pp. 639–647, 2018. DOI: 10.1039/C7LC01113J.

[7] K. A. Gopinathan, A. Mishra, *et al.*, "A microfluidic transistor for automatic control of liquids," *Nature*, vol. 622, no. 7984, pp. 735–741, 2023. DOI: 10.1038/s41586-023-06517-3.

[8] "Benchmarks — microfluidics." (), [Online]. Available: https://microfluidics.cs.ucr.edu/dmfb_static_simulator/benchmarks.

[9] W. H. Minhass, "System-level modeling and synthesis techniques for flow-based microfluidic very large scale integration biochips," Ph.D. dissertation, 2012.

[10] J. S. Marcus, W. F. Anderson, and S. R. Quake, "Microfluidic single-cell mRNA isolation and analysis," *Analytical Chemistry*, vol. 78, no. 9, pp. 3084–3089, 2006. DOI: 10.1021/ac0519460.

[11] C. Fang, Y. Wang, *et al.*, "Integrated microfluidic and imaging platform for a kinase activity radioassay to analyze minute patient cancer samples," *Cancer Research*, vol. 70, no. 21, pp. 8299–8308, 2010. DOI: 10.1158/0008-5472.CAN-10-0851.

[12] J. W. Hong, V. Studer, *et al.*, "A nanoliter-scale nucleic acid processor with parallel architecture," *Nature Biotechnology*, vol. 22, no. 4, pp. 435–439, 2004. DOI: 10.1038/nbt951.

[13] A. R. Wu, J. B. Hiatt, *et al.*, "Automated microfluidic chromatin immunoprecipitation from 2,000 cells," *Lab on a Chip*, vol. 9, no. 10, pp. 1365–1370, 2009. DOI: 10.1039/b819648f.

[14] B. Li, L. Li, *et al.*, "A smartphone controlled handheld microfluidic liquid handling system," *Lab on a Chip*, vol. 14, no. 20, pp. 4085–4092, 2014. DOI: 10.1039/C4LC00227J.

[15] T.-M. Tseng, M. Li, *et al.*, "Cloud columba: Accessible design automation platform for production and inspiration: Invited paper," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–6. DOI: 10.1109/ICCAD45719.2019.8942104.

[16] H. Huang and D. Densmore, "Fluigi: Microfluidic device synthesis for synthetic biology," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 11, no. 3, 26:1–26:19, 2015. DOI: 10.1145/2660773.

[17] B. Crites, C. Falzone, *et al.*, "Reducing microfluidic very large-scale integration (mVLSI) chip area by seam carving," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 10, pp. 2104–2116, 2021. DOI: 10.1109/TCAD.2020.3033499.