

# Digital and Analog Computing Paradigms in Printed Electronics

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik  
des Karlsruher Instituts für Technologie (KIT)

**genehmigte**

**Dissertation**

von

M.Sc.

**Dennis David Weller**

aus Saarbrücken, Deutschland

---

Tag der mündlichen Prüfung: 9. Dezember 2020

Referent: Prof. Dr. Mehdi Baradaran Tahoori  
Chair of Dependable Nano Computing  
Karlsruher Institut für Technologie

Korreferent: Prof. Dr. Jasmin Aghassi-Hagmann  
Fakultät für Elektrotechnik, Medizintechnik und Informatik  
Hochschule Offenburg

*To my family*

Hiermit erkläre ich an Eides statt, dass ich die von mir vorgelegte Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen - die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Karlsruhe, 25. März 2021  
Dennis D. Weller



# Acknowledgment

I wish to thank, first and foremost, my advisor Prof. Mehdi Tahoori for his professional guidance during my doctoral studies and who encouraged me to push the limits further. Without his persistent help, the goal of this project would not have been realized.

I also want to thank my second advisor Prof. Jasmin Aghassi-Hagmann, for her continuous support and for provisioning the tools and equipment I needed to conduct my research.

I would also like to sincerely thank Professor Michael Beigl from the TECO research group, for his insightful comments and suggestions.

I am additionally thankful to my PHD fellow Michael Hefenbrock, for his invaluable assistance that he provided during my study.

I am indebted to my many colleagues, Dr. Rajendra Bishnoi, Ahmet Erozan, Xiaowei Feng, Dr. Mohammad Golanbari, Farhan Rasheed, Alexander Scholz, for their practice contributions. Without their help, this work would not be resulted in that form.

Special Thanks also go to Dr. Gabriel Cadilha Marques and Felix Neuper, for introducing me into the fabrication of printing processes and measurement equipment, as well as all other colleagues from the Institute of Nanotechnology.

I'd also like to extend my gratitude to the research group from University of Illinois Urbana-Champaign, in particular Prof. Rakesh Kumar, Nathaniel Bleier, Mohammad Mubarik and Matthew Tomei.

I am also deeply thankful for the great support and love of my girlfriend Azadeh, my parents, sister and brother, for helping me to keep going on.

Finally, my thanks go to the Ministry of Science, Research and Arts of the state of Baden-Württemberg, which supported me financially in the form of the MERAGEM doctoral program.



# ABSTRACT

Due to the end of Moore's law, emerging materials and technologies have to be developed to saturate the innovation-driven IT market with new types of consumer electronics to allow penetration of currently inaccessible fields such as low-cost and flexible hardware, beyond the capabilities of existing silicon-based integrated circuits. In particular interesting are electronics for the fast-moving consumer goods market, which allows item-tagging, smart packaging or quality monitoring of disposables, where short shelf lifetime prevails. Other applications domains are anticipated such as flexible wearable devices or Internet of Things (IoT) infrastructures, which have stringent requirements on stretchability or non-toxicity.

To this end, additive manufacturing technologies such as printed electronics (PE) are considered which complement existing silicon-based electronics and provide very low production costs due to the simplification of the fabrication process, where functional and non-toxic materials are deposited on a wide range of substrates, including flexible plastic or paper carrier materials. Especially inkjet-printing is a promising candidate for future PE applications, as it enables on-site and on-demand printed hardware due to its mask-less fabrication process. However, as printed electronics technology is still in its very infancy, it is still questionable how circuits can be designed, which provide the required functionality to solve computational tasks for future applications and are at the same time manufacturable. As especially the feature sizes are several orders of magnitude higher than silicon-based electronics, and second either p- or n-type transistors are available, existing circuit designs from conventional electronics cannot be mapped to the PE domain. As a result, it is an open research question, what kind of computing paradigms are relevant for the development of circuit designs in this immature technology. The answer to this question will guide circuit designers in the future to determine how printed hardware can be developed for the novel market of consumer electronics.

For this purpose, this thesis explores different computing paradigms and circuit designs which are considered to be pertinent for future printed computing systems. The analysis is based on the recently developed electrolyte-gated transistors (EGTs) technology, which enables fabrication of low-voltage and inkjet-printed circuits using very low-cost material printers. As only simple logic gates were investigated in EGT-technology so far, the design space is further explored by consideration of memory elements, look-up tables, artificial neurons and decision trees. In particular the investigations of printed artificial neurons and decision trees contain already design flows for mapping large-scale machine learning classifiers to printed hardware.

The computing paradigms evaluated in this work range from digital, analog, neuromorphic and stochastic computing. Moreover, bespoke designs were analyzed, which leverage the customization capabilities of inkjet-printing to improve area usage and power consumption by hardwiring adjustable design parameters into the printed circuit. As the explored designs surpass the complexity of existing hardware solutions in EGT-technology, there is in principle

no guarantee that the circuits are manufacturable, especially the non-digital designs in this work. For this reason, EGT-based hardware prototypes were fabricated and characterized in terms of area usage, power consumption and performance. The measurement results can be used to extrapolate to large-scale circuits, even up to the application level. Moreover, as a proof-of-concept, the printed machine-learning classifiers in this thesis were tested and validated on popular benchmark datasets.

Overall, from the conducted experiments and evaluations in this thesis, several conclusions can be drawn. First, sequential operations for digital computing platforms can be performed by combining combinational logic with the proposed EGT-based storage elements. Next, analog and neuromorphic computing-based designs can be successfully deployed to realize low-complex and efficient machine learning classifiers. Also, stochastic computing neural networks have the benefit of reducing the high hardware footprint compared to conventional implementations. In general, bespoke and unconventional circuit designs are strongly encouraged in inkjet-printing technology to widen the applicability of printed hardware solutions.

It is believed that the results of this thesis will attract attention from material scientists, electrical engineers and computer scientists, which are actively working in the field of printed electronics. The investigated computing paradigms can be deployed in a meaningful way to design printed hardware for future application domains.



## ZUSAMMENFASSUNG

Da das Ende von Moore's Gesetz schon absehbar ist, müssen neue Wege gefunden werden um den innovationsgetriebenen IT-Markt mit neuartiger Elektronik zu sättigen. Durch den Einsatz von kostengünstiger Hardware mit flexiblem Formfaktor, welche auf neuartigen Materialien und Technologien beruhen, können neue Anwendungsbereiche erschlossen werden, welche über konventionelle siliziumbasierte Elektronik hinausgehen. Im Fokus sind hier insbesondere elektronische Systeme, welche es ermöglichen Konsumgüter für den täglichen Bedarf zu überwachen - z.B. im Zusammenhang einer Qualitätskontrolle - indem sie in das Produkt integriert werden als Teil einer intelligenten Verpackung und dadurch nur begrenzte Produktlebenszeit erfordern. Weitere vorhersehbare Anwendungsbereiche sind tragbare Elektronik oder Produkte für das "Internet der Dinge". Hier entstehen Systemanforderungen wie flexible, dehnbare Hardware unter Einsatz von ungiftigen Materialien.

Aus diesem Grund werden additive Technologien herangezogen, wie zum Beispiel gedruckte Elektronik, welche als komplementär zu siliziumbasierten Technologien betrachtet wird, da sie durch den simplen Herstellungsprozess sehr geringe Produktionskosten ermöglicht, und darüber hinaus auf ungiftigen und funktionalen Materialien basiert, welche auf flexible Plastik- oder Papiersubstrate aufgetragen werden können. Unter den verschiedenen Druckprozessen ist insbesondere der Tintenstrahldruck für zukünftige gedruckte Elektronikanwendungen interessant, da er eine Herstellung vor Ort und nach Bedarf ermöglicht auf Grund seines maskenlosen Druckprozesses. Da sich jedoch die Technologie der Tintenstrahl-druckbaren Elektronik in der Frühphasenentwicklung befindet, ist es fraglich ob Schaltungen für zukünftige Anwendungsfelder überhaupt entworfen werden können, beziehungsweise ob sie überhaupt herstellbar sind. Da die laterale Auflösung von Druckprozessen sich um mehrere Größenordnungen über siliziumbasierten Herstellungstechnologien befindet und des Weiteren entweder nur p- oder n-dotierte Transistoren verfügbar sind, können existierende Schaltungsentwürfe nicht direkt in die gedruckte Elektronik überführt werden. Dies führt zu der wissenschaftlichen Fragestellung, welche Rechenparadigmen überhaupt sinnvoll anwendbar sind im Bereich der gedruckten Elektronik. Die Beantwortung dieser Frage wird Schaltungsdesignern in der Zukunft helfen, erfolgreich gedruckte Schaltungen für den sich rasch entwickelnden Konsumgütermarkt zu entwerfen und zu produzieren.

Aus diesem Anlass exploriert diese Arbeit verschiedene Rechenparadigmen und Schaltungsentwürfe, welche als essenziell für zukünftige, gedruckte Systeme betrachtet werden. Die erfolgte Analyse beruht auf der recht jungen "Electrolyte-gated Transistor" (EGT) Technologie, welche auf einem kostengünstigen Tintenstrahldruckverfahren basiert und sehr geringe Betriebsspannungen ermöglicht. Da bisher nur einfache Logik-Gatter in der EGT-Technologie realisiert wurden, wird in dieser Arbeit der Entwurfsraum weiter exploriert, durch die Entwicklung von gedruckten Speicherbausteinen, Lookup Tabellen, künstliche Neuronen und Entscheidungsbäume. Besonders bei dem künstlichen Neuron und den Entscheidungsbäumen wird Bezug auf Hardware-Implementierungen von Algorithmen des maschinellen Lernens gemacht und die Skalierung der Schaltungen auf die Anwendungsebene aufgezeigt.

Die Rechenparadigmen, welche in dieser Arbeit evaluiert wurden, reichen von digitalen, analogen, neuromorphen Berechnungen bis zu stochastischen Verfahren. Zusätzlich wurden

individuell anpassbare Schaltungsentwürfe untersucht, welche durch das Tintenstrahldruckverfahren ermöglicht werden und zu substanziellen Verbesserungen bezüglich des Flächenbedarfs, Leistungsverbrauch und Schaltungslatenzen führen, indem variable Entwurfsparameter in die Schaltung fest verdrahtet werden. Da die explorierten Schaltungen die Komplexität von bisher hergestellter, gedruckter Hardware weit übertreffen, ist es prinzipiell nicht automatisch garantiert, dass sie herstellbar sind, was insbesondere die nicht-digitalen Schaltungen betrifft. Aus diesem Grund wurden in dieser Arbeit EGT-basierte Hardware-Prototypen hergestellt und bezüglich Flächenbedarf, Leistungsverbrauch und Latenz charakterisiert. Die Messergebnisse können verwendet werden, um eine Extrapolation auf komplexere anwendungsbezogenere Schaltungsentwürfe durchzuführen. In diesem Zusammenhang wurden Validierungen von den entwickelten Hardware-Implementierungen von Algorithmen des maschinellen Lernens durchgeführt, um einen Wirksamkeitsnachweis zu erhalten.

Die Ergebnisse dieser Thesis führen zu mehreren Schlussfolgerungen. Zum ersten kann gefolgert werden, dass die sequentielle Verarbeitung von Algorithmen in gedruckter EGT-basierter Hardware prinzipiell möglich ist, da, wie in dieser Arbeit dargestellt wird, neben kombinatorischen Schaltungen auch Speicherbausteine implementiert werden können. Letzteres wurde experimentell validiert. Des Weiteren können analoge und neuromorphe Rechenparadigmen sinnvoll eingesetzt werden, um gedruckte Hardware für maschinelles Lernen zu realisieren, um gegenüber konventionellen Methoden die Komplexität von Schaltungsentwürfen erheblich zu minimieren, welches schlussendlich zu einer höheren Produktionsausbeute im Herstellungsprozess führt. Ebenso können neuronale Netzwerkarchitekturen, welche auf Stochastic Computing basieren, zur Reduzierung des Hardwareumfangs gegenüber konventionellen Implementierungen verwendet werden. Letztlich kann geschlussfolgert werden, dass durch den Tintenstrahldruckprozess Schaltungsentwürfe bezüglich Kundenwünschen während der Herstellung individuell angepasst werden können, um die Anwendbarkeit von gedruckter Hardware generell zu erhöhen, da auch hier geringerer Hardwareaufwand im Vergleich zu konventionellen Schaltungsentwürfen erreicht wird.

Es wird antizipiert, dass die in dieser Thesis vorgestellten Forschungsergebnisse relevant sind für Informatiker, Elektrotechniker und Materialwissenschaftler, welche aktiv im Bereich der druckbaren Elektronik arbeiten. Die untersuchten Rechenparadigmen und ihr Einfluss auf Verhalten und wichtige Charakteristiken gedruckter Hardware geben Einblicke darüber, wie gedruckte Schaltungen in der Zukunft effizient umgesetzt werden können, um neuartige auf Druckverfahren-basierte Produkte im Elektronikbereich zu ermöglichen.

# Table of Contents

<b>ACKNOWLEDGMENT</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Problem Statement and Thesis Goal . . . . .	2
1.2. Thesis Contributions . . . . .	3
1.2.1. Inkjet-Printed Digital Circuits . . . . .	4
1.2.2. Inkjet-Printed Neuromorphic Architectures . . . . .	4
1.2.3. Inkjet-Printed Analog Circuits . . . . .	5
1.3. Summary . . . . .	5
1.4. Outline . . . . .	6
<b>2. Background and State of the Art</b>	<b>7</b>
2.1. Printed Electronics - Overview . . . . .	7
2.2. Inkjet-Printing . . . . .	11
2.3. Electrolyte-Gated Transistors . . . . .	12
2.4. EGT-based Circuit Fabrication . . . . .	12
2.5. Exploration of New Designs and Computing Paradigms . . . . .	13
<b>3. Inkjet-Printed Digital Circuits</b>	<b>15</b>
3.1. Inkjet-Printed SR-latch . . . . .	15
3.1.1. Latch Design . . . . .	15
3.1.2. Simulation Environment . . . . .	15
3.1.3. Fabrication . . . . .	16
3.1.4. Measurement Setup . . . . .	16
3.1.5. Results . . . . .	18
3.1.6. Summary . . . . .	18
3.2. Inkjet-Printed Lookup-Table . . . . .	19
3.2.1. Proposed Printed Lookup Table Design . . . . .	19
3.2.2. Hardware Prototype . . . . .	22
3.2.3. Summary . . . . .	24
<b>4. Inkjet-Printed Neuromorphic Architectures</b>	<b>27</b>
4.1. Inkjet-Printed Analog Artificial Neural Network . . . . .	27
4.1.1. Artificial Neural Networks (ANN) . . . . .	28
4.1.2. Printed Programmable Neuron . . . . .	28

4.1.3.	Improvements on the Proposed ANN design . . . . .	33
4.1.4.	Proposed ANN Hardware Architecture . . . . .	38
4.1.5.	Training of Printed ANN . . . . .	40
4.1.6.	Benchmark Results . . . . .	43
4.1.7.	Summary . . . . .	43
4.2.	Inkjet-Printed Stochastic Computing Neural Networks . . . . .	45
4.2.1.	Stochastic Computing . . . . .	45
4.2.2.	Related Work on SC-based NNs . . . . .	47
4.2.3.	Proposed SC Designs for PE . . . . .	48
4.2.4.	Simulation Results . . . . .	53
4.2.5.	Summary . . . . .	56
<b>5.</b>	<b>Inkjet-Printed Analog Circuits</b>	<b>59</b>
5.1.	Inkjet-Printed Decision Tree . . . . .	59
5.1.1.	Conventional Digital Binary Decision Trees . . . . .	60
5.1.2.	Bespoke Digital Binary Decision Trees . . . . .	61
5.1.3.	Analog Binary Decision Trees . . . . .	63
5.1.4.	Training . . . . .	65
5.1.5.	Summary . . . . .	67
5.2.	Inkjet-Printed Analog Read-Only Memory . . . . .	69
5.2.1.	Summary . . . . .	70
<b>6.</b>	<b>Summary, Conclusion and Outlook</b>	<b>73</b>
6.1.	Summary . . . . .	73
6.2.	Conclusions . . . . .	75
6.3.	Outlook . . . . .	77
	<b>Appendices</b>	<b>79</b>
<b>A.</b>	<b>Fabrication of EGTs</b>	<b>81</b>
A.1.	Printing Steps . . . . .	81
A.2.	Ink Preparation . . . . .	81
<b>B.</b>	<b>Resistance to Neural Network Weight Calculation</b>	<b>83</b>
	<b>Bibliography</b>	<b>87</b>

# List of Figures

1.1. Photo of organic sensor . . . . .	2
2.1. Subtractive vs additive fabrication . . . . .	8
2.2. Printed vs silicon-based electronics . . . . .	9
2.3. Fabrication processes in PE . . . . .	10
2.4. Resolution and throughput of fabrication processes in PE . . . . .	11
2.5. Desktop functional inkjet-printer - Dimatix DMP-2850 . . . . .	12
2.6. EGT material stack and photo . . . . .	13
2.7. EGT printing steps . . . . .	14
3.1. Equivalent circuit of the SR-latch . . . . .	16
3.2. Photo of SR-latch . . . . .	17
3.3. Transient measurements of the printed SR-latch. . . . .	17
3.4. Schematic and truth table of the proposed 1-input LUT . . . . .	20
3.5. Schematic of XNOR programmed LUT . . . . .	21
3.6. Photo of printed LUT-based XNOR . . . . .	23
3.7. Measured waveforms of the printed LUT2 configured as an XNOR . . . . .	23
3.8. Photo of printed LUT-based XOR . . . . .	24
3.9. Measured waveforms of the printed LUT2 configured as an XOR . . . . .	24
3.10. Photo of printed LUT-based AND . . . . .	25
3.11. Measured waveforms of the printed LUT2 configured as an AND . . . . .	25
4.1. Schematic, photo and measurements of MAC circuit . . . . .	31
4.2. Measured waveforms of the printed piece-wise linear unit (pPLU) . . . . .	33
4.3. Schematic, photo and measurements of printed 2-input neuron . . . . .	34
4.4. Schematic, photo and measurements of inv circuit . . . . .	37
4.5. Schematic, photo and measurements of ptanh circuit . . . . .	39
4.6. Printed ANN - from architecture to circuit level . . . . .	41
4.7. Implementation of artificial neuron components using digital computing vs stochastic computing . . . . .	49

*LIST OF FIGURES*

4.8. Schematic and photo of analog stochastic number generator . . . . .	50
4.9. Simulation of analog stochastic number generator . . . . .	51
4.10. Schematic of analog SC-based activation function . . . . .	52
4.11. Simulation of analog SC-based activation function . . . . .	53
4.12. Illustration of implementation of mixed-signal SC-based neuron . . . . .	54
4.13. Bar chart - comparison between digital ANN and SC-NN . . . . .	56
5.1. Illustrative example of binary decision tree . . . . .	60
5.2. Design flow of bespoke digital decision tree . . . . .	62
5.3. Measured waveforms of the printed bespoke digital decision tree with 2-bit precision	63
5.4. Schematic and layout of analog decision tree . . . . .	66
5.5. Schematic, photo and measurements of 4×1 ROM . . . . .	71
6.1. Comparison of latch and ROM . . . . .	74
6.2. Comparison of different LUT implementations . . . . .	74
6.3. Comparison of neuron implemented in digital, analog and stochastic computing	75
6.4. Comparison different computing paradigms for decision trees . . . . .	76

# List of Tables

3.1. SR-Latch truth table . . . . .	16
3.2. Comparison of LUT2 implementations based on EGT technology . . . . .	22
4.1. Design parameters and measurements of fabricated ANN components . . . . .	32
4.2. Design parameters for 2-input neuron fabrication . . . . .	33
4.3. Comparison of digital and analog ANN . . . . .	40
4.4. ANN inference results on benchmark datasets . . . . .	44
4.5. ANN components in digital vs SC computing . . . . .	55
4.6. Comparison of ANN neuron in digital vs SC computing . . . . .	55
4.7. Inference results of SC-NN on benchmark datasets . . . . .	57
5.1. Comparison of conventional digital vs parallel decision trees . . . . .	61
5.2. Inference results of decision trees on benchmark datasets . . . . .	67
6.1. Comparison of computing paradigms of the fabricated hardware prototypes in this work . . . . .	73





# 1. Introduction

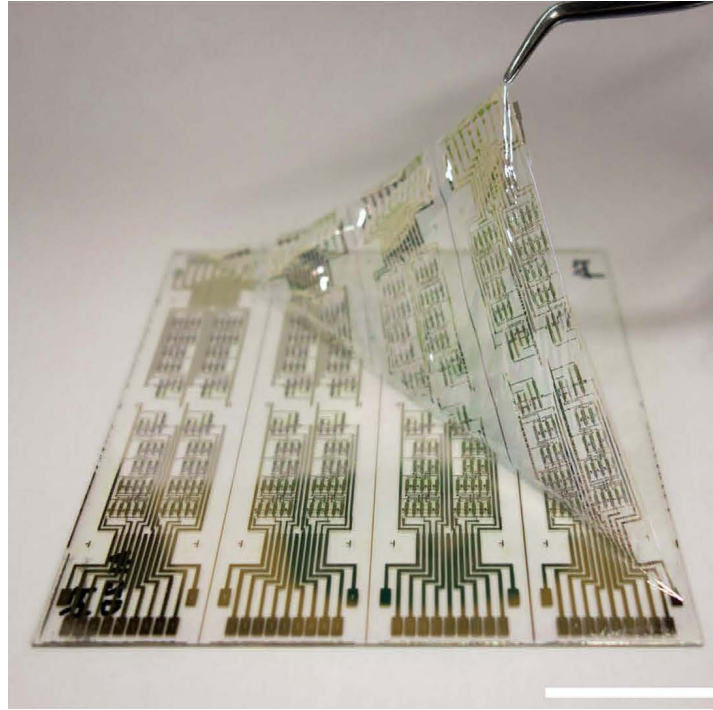
Moore's law, which states that transistor count in integrated circuits doubles about every two years, is predicted to come to an end in the next years after reaching the limits of miniaturization at the size of atoms [1]. As a result, the innovation-driven IT industry is searching for alternative technologies to improve electronics for the large customer market [2]. An enormous research space is explored beyond the established technologies [2], to allow the penetration of application domains, which were still untouched by conventional hardware due to limitations of silicon-based technology such as high production costs and restrictions on conformity. It is anticipated, that for the future consumer goods market, even consumables are required without integrated computing devices, such as: identification and tracking [3], brand authentication [4] or quality monitoring [5]. Finally, several domains demand also for stretchability, porosity, non-toxicity that silicon-based systems cannot meet [6].

Emerging technologies are already deployed to enable soft robotics [7], soft sensors [8, 9], flexible devices (see Figure 1.1) or Internet of things (IoT) infrastructures [10]. Low-cost point-of-use fabrication techniques are under development to reduce the time-to-market of products, which are part of the fast-moving consumer goods market [11]. In this respect, printed electronics (PE) technology can be leveraged to lower the production costs as it relies on very simple fabrication techniques and can totally avoid costly subtractive processes, commonly used in silicon electronics. Only small printers are required compared to expensive silicon foundries [12] for even older technology nodes. As a result, portable products [13] with short production time [14] and reduced production costs [15] are enabled. Due to the additive process, a wide range of carrier materials is supported to achieve conformable [16] and non-toxic [17] hardware. It is expected that the PE market will grow from \$24B in 2017 to about \$200B in 2027 [14]. Although the performance compared to silicon-based electronics is several orders of magnitude lower, several applications are already targeted by PE, such as flexible sensors and sensor processing solutions for the IoT [18] and the wearable electronic market [19], applications which would be simply out of reach for energy-hungry and complex full-scale silicon-based systems on chips [20].

Among the different fabrication processes in PE, especially inkjet-printing - which is the target technology in this work - gained a lot of attraction as it enables on-demand and on-site fabrication due to its mask-less fabrication process. Costs for inkjet-printing equipment is lower compared to roll-to-roll printing processes and thus already used widely in offices, homes and factories [23]. Several functional materials were proposed for inkjet-printing processes recently [24]. Electrolyte-gated transistors (EGTs) were presented [25], which operate below two volts, making it a suitable choice for portable and battery-driven electronics for pervasive computing tasks.

However, as EGT technology emerged most recently and is still in its very infancy, it is an open research question how future applications can benefit from it. It is well known, that future computing paradigms have to be based on the underlying physical process [26], thus

## 1. Introduction



**Figure 1.1.:** *Photograph of flexible organic CMOS logic circuit [21]. Reprint from [22] under license CC*

this work explores several computing paradigms, ranging from digital over neuromorphic to analog computing.

### 1.1. Problem Statement and Thesis Goal

For realizing electronics in EGT-technology for deployment in the aforementioned future application domains, still several research questions exist:

- As investigations on EGT-based inkjet-printing technology was only limited to single logic gates so far, a more application-specific analysis has to be carried out by focusing on more complex circuit designs.
- As it is foreseen that PE will play a major role for direct sensor processing tasks [23], it is of great interest how machine learning algorithms can be mapped to printed hardware for solving classification problems. Until now, no comprehensive analysis was conducted on printed machine learning classifiers.
- The benefits of inkjet-printing compared with screen- or roll-to-roll printing haven't been evaluated so far. However, it is assumed that due to the point-of-use and customized fabrication process of inkjet-printing, new circuit designs will emerge, not feasible or economical by non-inkjet-printing methods.
- As it is proven by the previous EGT-based logic gates implementations, that digital designs can be manufactured, it is particularly interesting how other computing paradigms can be applied to EGT-technology and whether they can improve circuit characteristics.

These research questions are addressed in this thesis as follows:

- The previously explored design space in inkjet-printed EGT technology, which was limited to single logic gates, is extended by more complex printed circuits with reference to potential target applications.
- It is investigated, how popular machine learning algorithms can be efficiently mapped to inkjet-printed hardware.
- The customization feature of inkjet-printing is leveraged for the majority of the analysed circuit designs to emphasize benefits compared to non-inkjet printing methods.
- Besides digital computing, several other computing paradigms which are of high relevance for future PE applications were evaluated, such as neuromorphic, analog and stochastic computing.

The thesis contributions are described in more detail in the following section.

## 1.2. Thesis Contributions

The thesis contributions are subdivided into three categories: digital, neuromorphic and analog computing paradigms. The chapter about inkjet-printed digital circuits is related to the research question how the customization capability of inkjet-printing can improve circuit characteristics compared to screen- and roll-to-roll processes. Moreover, an inkjet-printed digital memory element is proposed, which validates that also sequential operations can be performed in EGT-technology. In the neuromorphic computing chapter, it is evaluated how a machine learning classifier can be efficiently mapped to EGT-technology by using printed neuromorphic computing systems as well as stochastic computing based neural networks. Also in this chapter, inkjet-printed customized designs were analyzed. Finally, in the analog computing chapter, decision tree-based classifiers are proposed. The advantage of using analog computing for reduction in circuit complexity is evaluated. Besides an efficient implementation of a machine learning classifier, also an analog read-only-memory is presented.

As inkjet-printed circuit designs are expensive to be evaluated due to the immature fabrication process in EGT-technology, for each explored computing paradigm only a few design points were investigated, which have however high relevance to address the aforementioned open research questions in this work, and are at the same time manufacturable. As several technology-related limitations exist in EGT-technology, such as high intrinsic variations due to non-determinism of droplet printing [23, 27], it is in general not ensured that newly developed circuit designs can be fabricated. Thus printed hardware prototypes were included in this work as a proof of concept. Although the chosen representative circuit designs might appear to be simple and low-complex from the system-level perspective, they still provide valuable insights as important parameters such as area usage, power consumption and performance can be extrapolated to larger designs. Nevertheless, for validation of the printed machine learning classifier hardware, also high-level simulations were performed for validating the hypothesis that classification problems can be solved.

The main contributions of this PhD thesis are summarized in the following in more detail.

### 1.2.1. Inkjet-Printed Digital Circuits

A digital and sequential computing paradigm was explored by design and fabrication of a 1-bit storage element, the SR-latch. Due to the absence of p-type transistors, mapping of existing CMOS-based designs from the silicon-based domain to PE is not feasible. Moreover, as the n-type EGTs are subject to very high process variations, circuits with less EGTs are highly favourable. In [28], an inkjet-printed SR-latch was proposed, which requires only four EGTs in the pull-down network, and two resistors in the pull-up network. The fabricated latch operated at very low supply voltages of only 0.6V, making it compatible with prospective energy harvester systems. This contribution was published in [28]. The most part of the respective Section 3.1 are identical with [28].

Within the digital computing domain, also a one-time programmable lookup table (LUT) was investigated. While LUTs are commonly used for general-purpose computing and field programmable gate arrays (FPGAs), the concept can be leveraged to enable the customization capabilities of inkjet-printing technology. To this end, a customization technique was developed, which leads to less transistors compared to conventional designs. Consequently, the designed 2-bit LUT, requires only five EGTs, and is thus more reliable compared to conventional boolean-logic based implementations (which require eight EGTs). The LUT can be programmed by printing conductive connections between circuit elements which were separated before to obtain a bespoke design. By cascading multiple LUTs, any boolean logic can be realized. The printed LUT is a suitable choice for systems in PE, as it can be used to mask hardware failures through its rerouting ability and thus increases chip yield. This contribution was published in [29]. Major parts of the respective Section 3.2 are identical with [29].

### 1.2.2. Inkjet-Printed Neuromorphic Architectures

In order to perform efficient near-sensor computations, integration of printed build-in machine learning classifiers are advantageous. However, due to the large feature sizes, high-process variations and thus requirements on low-device count, digital designs might be sub-optimal for several applications. In order to increase the functional density of printed circuits, neuromorphic computing paradigms are explored, which are inspired by the behavior and structure of the biological brain, resulting in substantial improvements in terms of area usage and power consumption. Moreover, neural network architectures have the capability of being intrinsically fault tolerant. To this end, neuromorphic designs were explored in inkjet-printing technology. An inkjet-printed and programmable neuron was designed, which is a fundamental building block for larger artificial neural networks (ANN). The neuron design consists of the two neural network core operations: multiply-accumulate (MAC) operation and non-linear activation function. The printed neuron consists of only one EGT and additionally one printed resistor per neuron input, resulting in a very small hardware footprint. As it was demonstrated with the hardware prototype, the functionality of the printed neuron can be programmed by printing different-sized resistors at the crossbar-interconnects of the MAC circuit. The neuron design was further extended by proposing circuits for negative ANN weights as well as tanh-like activation functions. It was validated that classification problems can be solved with the proposed printed neuromorphic hardware architecture. Parts of this contribution were published in [30] or have been submitted at [31]. Thus, the published contents are identical with the respective Section 4.1.

Another computing paradigm, which was explored for the implementation of neural algorithms is Stochastic Computing (SC). For efficient neural network implementations, in the SC domain expensive multipliers and adder circuits are replaced by single logic gates such as XNOR gates and multiplexers. This allows for very small hardware usage and previously infeasible design architectures become manufacturable. However, as signal generators and activation functions in SC-based ANNs are still expensive to be produced, analog circuits were derived as an replacement. The benefits of the proposed mixed-signal SC-based ANN are evaluated and the concept was validated on popular benchmark datasets. This contribution was published in [32] (in press). Major parts of the respective Section 4.2 are identical with [32] (in press).

### 1.2.3. Inkjet-Printed Analog Circuits

Besides neural networks, another possibility to realize machine learning classifiers is the utilization of decision trees. In this regard, a digital tree-based classifier was developed, which is capable of solving classification tasks. The advantage of the proposed decision tree design was the possibility to implement it as a bespoke circuit, in order to reduce the device count substantially. Following the analog computing paradigm, also an analog tree-based classifier was designed, which is 5x faster, requires 582x less area and 178x less power consumption compared to the digital counterpart. The functionality of the printed decision tree is defined by customization of printed resistors in the circuit. Evaluations on benchmark datasets have indicated that the proposed tree-architecture can solve classification problems. This contribution was published in [33]. Major parts of the respective Section 5.1 are identical with [33].

Finally, also an analog implementation of a read-only memory (ROM) was designed and fabricated. The one-time programmable ROM element consists of a resistive crossbar architecture with EGT-based decoder logic. Information is encoded with different resistance states of the crossbar resistors. Per ROM address, two bit of information could be encoded, leading to very small area usage per stored information. This contribution was published in [33]. Major parts of the respective Section 5.1 are identical with [33].

## 1.3. Summary

In this thesis, for the first time, several computing paradigms for the emerging EGT-based inkjet-printing technology were evaluated. It is believed that this work paves the way for the realization of future printed computing systems and help circuit designers to decide what kind of computing paradigms are suitable given a specific application and use case. Pertinent parameters such as circuit delay, power consumption or area usage are provided and allow for extrapolation to large-scale printed hardware or even complete printed systems which are beyond the presented circuits in this work. Whenever possible, the proposed printed hardware was validated on the system level by deployment of architecture-level simulations. For the machine learning classifiers, popular benchmark dataset were utilized for validating the circuit functionality and design concepts.

## 1.4. Outline

The remaining thesis is organized as follows:

- Chapter 2 provides the background knowledge about PE in general as well as EGT-based technology.
- Chapter 3 contains circuit designs in PE implemented by following the digital computing paradigms. It contains the evaluation of a printed 1-bit storage element as well as a 2-bit LUT.
- Chapter 4 discusses printed neuromorphic computing hardware, which is about the implementation and evaluation of an artificial neural network as well as stochastic-computing neural networks in PE.
- Chapter 5 evaluates in the first part decision trees in PE, which includes conventional and bespoke digital as well as analog realizations of binary decision trees. In the second part, a bespoke and analog 8-bit ROM is presented and discussed.
- Chapter 6 summarizes and concludes this thesis.

## 2. Background and State of the Art

In this chapter, the required background knowledge about printed electronics (PE) is provided. An overview of printed electronics is given with its different manufacturing processes including inkjet-printing. The proposed target technology, which is based on inkjet-printed electrolyte-gated transistors (EGTs), is introduced in this chapter with reference to previous achievements in EGT technology. Finally, the unresolved research questions are discussed and how this thesis advances the state of the art.

### 2.1. Printed Electronics - Overview

With the recent advance of new functional materials, next-generation electronics are facilitated, which are beyond the capability of silicon-based VLSI technology limited to bulky substrates with associated high production costs. In this regard, PE is a key enabler technology, which paves the way for future applications such as ultra-low-cost [34], flexible [35] and large-area [36] computing systems. Compared to lithography processes for wafer-scale silicon-based ICs, fabrication processes in PE are highly simplified as functional inks are directly deposited on a wide range of substrates, including flexible carrier materials [23, 35, 36, 37]. While some printing techniques are based on purely additive manufacturing steps, others deploy both additive and subtractive processes, as illustrated in Figure 2.1. Similar to silicon technology, subtractive processes in PE involve the deposition and development of photoresists with a subsequent etching step. As a result, subtractive methods, which require expensive infrastructure and equipment [38], are more expensive than additive processes. On the other hand, fully additive printing is only based on several simple deposition steps, where functional inks are printed layer-by-layer to realize active (transistors) and passive (resistors, capacitors) devices.

PE-based computing systems can be fabricated with very low per-unit-area costs. However, as the large feature sizes in PE are in the range of micrometers, PE circuits have low performance and small functional densities compared to silicon technology where feature size is in the nanometer range. An illustrative comparison between PE and silicon-based electronics is given in Figure 2.2. As none of these technologies are Pareto-optimal regarding performance and costs, PE is considered as a complementary solution to the silicon technology, which motivates the deployment of hybrid computing systems, where low-cost printed electronic circuits are combined with high-performance silicon-based components [10].

Similar to color printing, PE fabrication processes are categorized into different groups such as jet-printing [39, 40], screen printing [41, 42] or roll-to-roll [40, 43] processes. These printing processes are illustrated in Figure 2.3 and briefly described in the following. A more comprehensive discussion can be found in [23].

Jetting of functional inks in inkjet printers are controlled by a CAD software, which converts the digital pattern into voltage-pulsed jetting signals. These voltage pulses are applied to

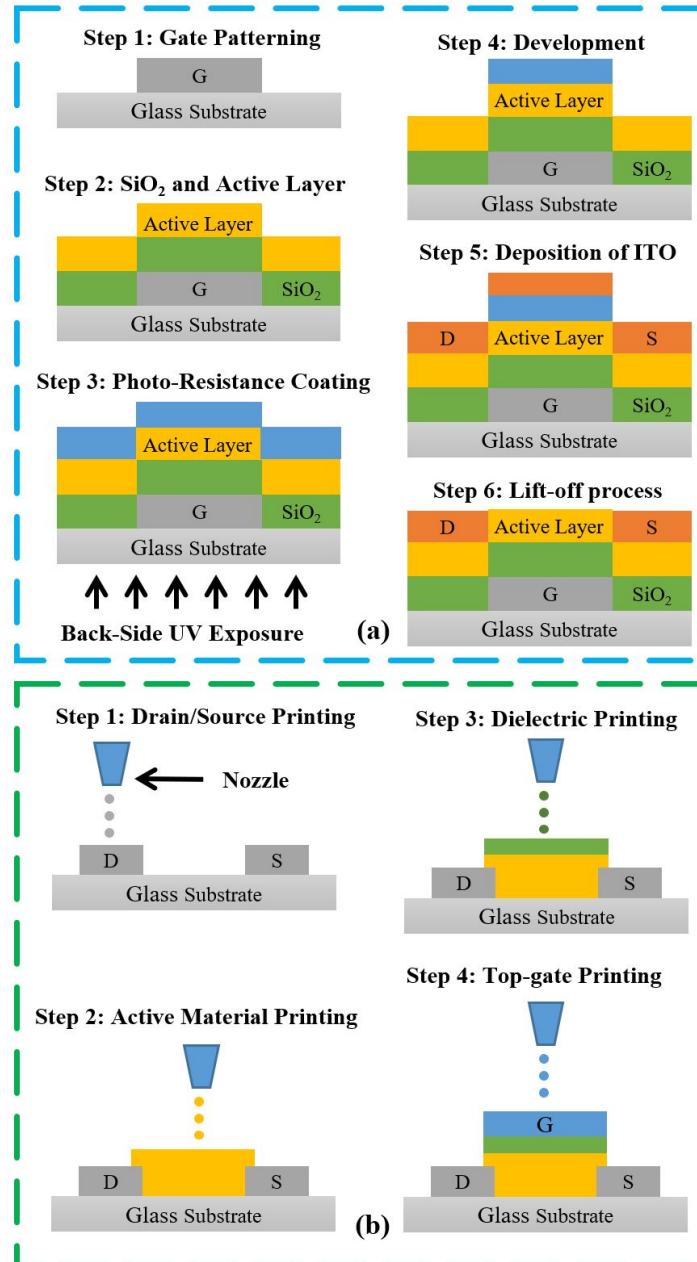
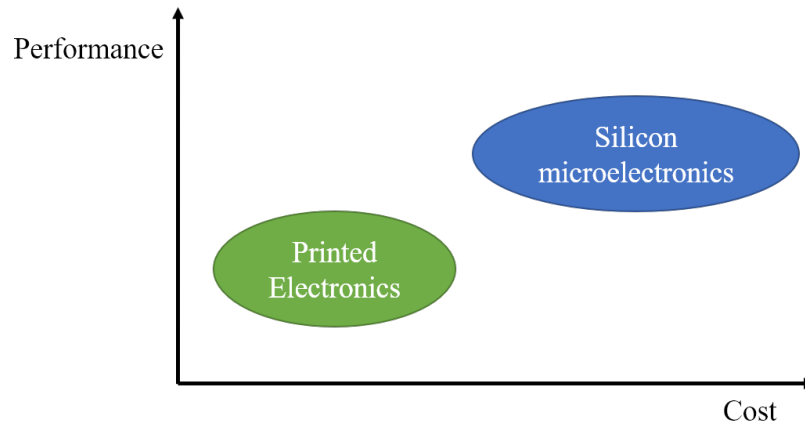


Figure 2.1.: (a) Subtractive-based printing process (b) Fully additive inkjet-printed process





**Figure 2.2.:** Comparison of printed electronics with silicon-based electronics in terms of performance and costs [23]

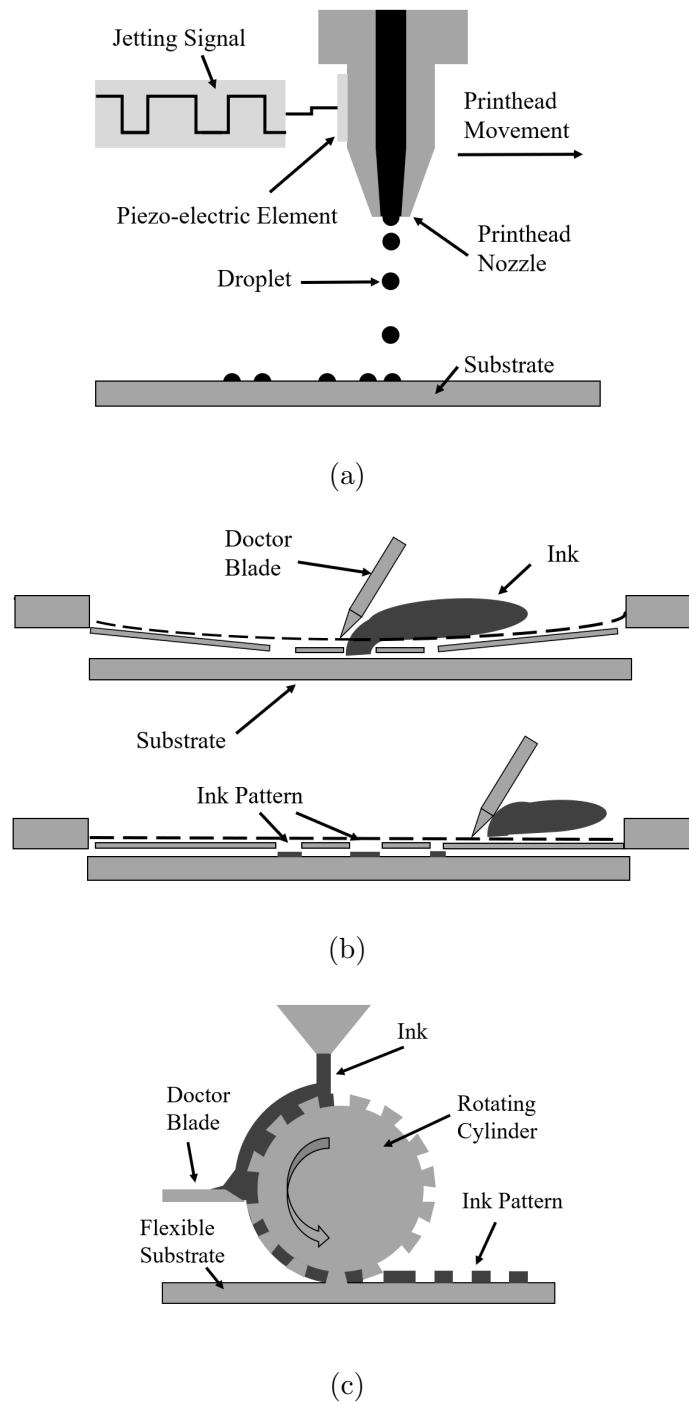
piezo-electric elements, which change the volume of the ink chamber in the inkjet printhead, and droplets are formed at the nozzle which move downwards to the substrate (see Figure 2.3a). By controlling the jetting phases and lateral movement of the printhead, printed patterns are deposited to the substrate. An advantage of inkjet printing is that low viscosity inks can be deployed and no contact is made between the printing equipment and substrate, thus enabling support for a wide range of carrier materials, including rough and flexible substrates. In addition, due to the digital and on-the-fly printing process, no master plate is required and the fabrication costs are reduced [23].

Screen printing, on the other hand, refers to an analog printing technique. In screen printing, a fabric mesh is produced by optical lithography which contains a layer of opaque and transparent pattern. After the mesh is covered with ink, a blade is moved over the mesh and presses the ink through the mesh to the substrate (see Figure 2.3b). Due to the simple fabrication, screen printing is widely used in industry. While costs for both equipment and ink are lower compared to other printing methods, printing resolution is low. Moreover, functional inks must be very viscous, which leads to a decrease in the electrical properties [23].

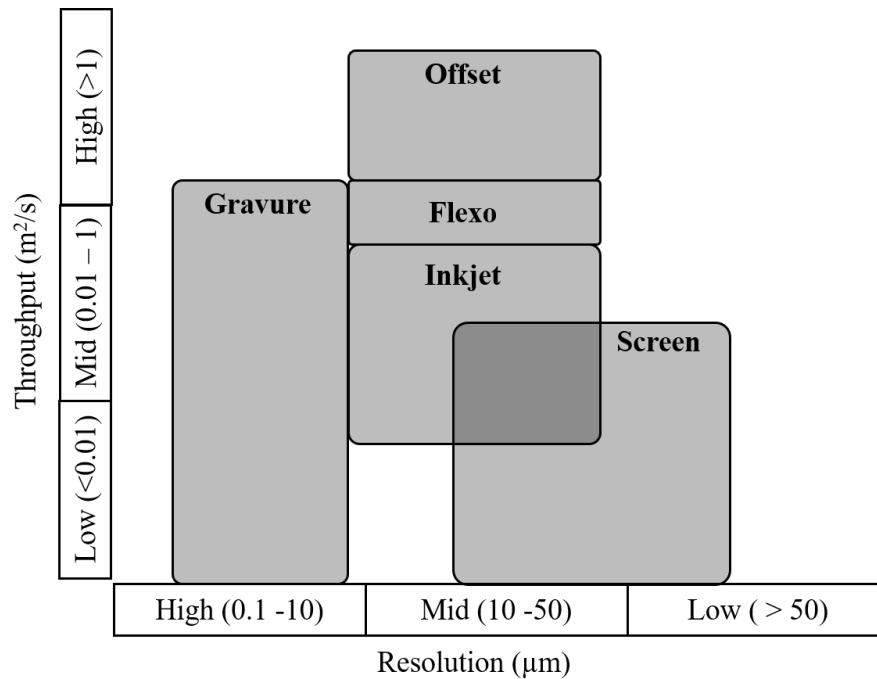
Also, gravure printing denotes an analog printing process. Before the printing step, a master plate - or also called gravure plate - must be manufactured by using laser engraving to obtain the printing pattern. After gravure plate fabrication, in the first step, the ink is deposited to the plate using an ink supply, where a doctor blade removes the excess of ink in a second step. The plate is then pressed to the substrate during a rotational movement, and printed patterns are produced dependent on the dot matrix of the gravure cylinder (see Figure 2.3c). Gravure printing enables the use of low-viscosity inks and, moreover, provides good control over total ink consumption. Millions of prints can be achieved with gravure printing without changing the gravure plate. However, due to the high costs for gravure plate fabrication, only the production of large batch sizes is economical [23].

While all of these printing methods refer to an additive manufacturing process, they differ in terms of printing resolution and production throughput [14] as it can be obtained from Figure 2.4. Gravure printing provides the highest printing resolution in the sub-10 $\mu\text{m}$  range. Also, maximum throughput is the highest and comparable to other roll-to-roll processes such as Flexo- and Offset-printing [14]. Both screen and inkjet printing have typically feature sizes

2. Background and State of the Art



**Figure 2.3.:** Comparison of printed electronics processes: a) inkjet-printing b) screen printing c) gravure printing



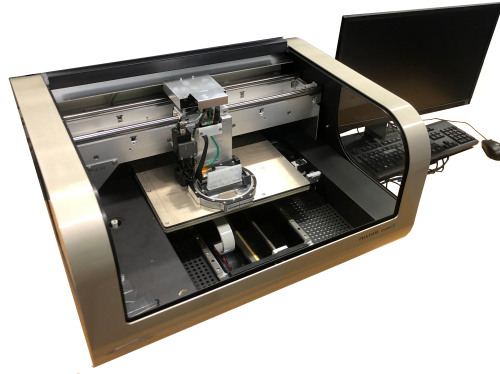
**Figure 2.4.:** Comparison of printed electronics processes in terms of printing resolution and throughput [14]

above 10µm with low-to-medium volume fabrication possibilities. In terms of carrier materials, all processes can be used to print on plastic or paper, while inkjet-printing and screen printing also support glass substrates [24].

## 2.2. Inkjet-Printing

Among the different manufacturing processes, inkjet printing, which is the targeted technology in this work - has the advantage of on-demand and on-site fabrication due to its mask-less fabrication process. As non-recurring engineering costs are very low, inkjet-printed designs can be tailored to the application and produced in low batch sizes using customized designs, generated on-the-fly. Such point-of-use and on-demand fabrication is required in many extremely low-cost and low-to-medium volume applications [23]. Due to the fab-in-the-box concept, customization of circuit designs can substantially improve hardware footprint, performance and power consumption [33], enable bespoke circuit designs which are not feasible with screen- or roll-to-roll printing and widen the applicability of printed electronics.

Such low-cost material printers (see Figure 2.5) are already widely used in offices, homes and factories [23] and cost only about 50,000\$. Although development and optimization of functional inks which match the rheological requirements of such inkjet printers is still in progress, most recently, fabrication of fully inkjet-printed transistors were presented [44]. These transistors are built from three basic materials, namely printable semiconductors, dielectrics and conductive films. Based on these materials, inkjet-printed implementation of logic gates on paper [45] were presented as well as printed diode-based organic rectifiers for near-field communication [46], which is interesting for future smart packaging solutions. Moreover, Digital-to-Analog



**Figure 2.5.:** *Desktop functional inkjet-printer - Dimatix DMP-2850*

converters [47], memories [48] and amplifiers [38, 49] were proposed.

The deployed printed circuits differ in terms of printing steps, performance and supply voltages, which ranges from several Volts up to hundreds of Volts [14]. However, the latter can lead to technical limitations, as printed circuits are anticipated to be driven by low-voltage printed batteries and energy harvesters. In the following, electrolyte-gated transistor technology is introduced, which is expected to be a promising candidate for realizing low-cost and low-voltage printed computing systems in the future.

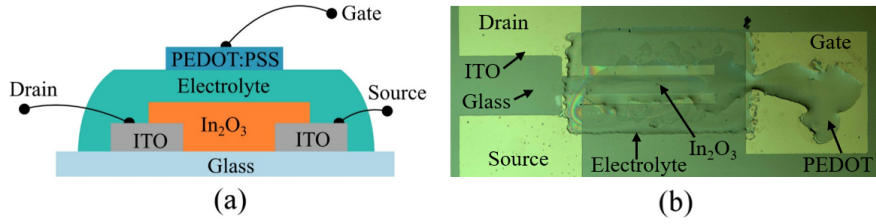
### 2.3. Electrolyte-Gated Transistors

Inorganic n-type electrolyte-gated field-effect transistors (EGT) were proposed in the past. They consist of inkjet-printed semiconductor materials, electrolytes as a dielectric substitute, and inkjet-printed conductive ink. The material stack of an EGT is depicted in Figure 2.6a. The three materials are deposited layerwise, starting with the semiconductor channel ( $\text{In}_2(\text{NO}_3)_3$  precursor ink, after annealing becomes  $\text{In}_2\text{O}_3$ ) printed on top of a passive conductive structure of sputtered indium tin oxide (ITO) on glass substrate, then printing a composite solid polymer electrolyte (CSPE) as well as a conductive material for the transistor top-gate contact [50] in the last step. A top view of a printed EGT is provided in Figure 2.6b.

Due to the high gate-capacitance of the electrolyte/semiconductor interface and low transistor threshold voltages ( $<200\text{mV}$ ), EGTs can operate below 1V [50]. This is in particular attractive for low-voltage and portable electronics.

### 2.4. EGT-based Circuit Fabrication

The inkjet printer deployed in this work for fabrication of EGT-based circuits is depicted in Figure 2.5, and it was used for deposition of all three functional materials required to print EGTs. The printing steps are illustrated in Figure 2.7 and are discussed in more detail in the following.



**Figure 2.6.:** *a) Side view of a printed EGT b) Top view of a fabricated EGT taken from a microscope photo*

The circuits are printed on a  $20\text{mm} \times 20\text{mm}$  glass substrate, which was sputtered by Sn-doped ITO. The passive components like resistors, electrodes, and interconnects were produced by structuring of the ITO substrate using eBeam lithography with poly(methyl methacrylate) (PMMA) as a photoresist. As an alternative, the ITO-substrate was structured by laser ablation.

Afterwards, EGT printing was performed by first inkjet-printing of the precursor  $\text{In}_2(\text{NO}_3)_3$  between the drain and source electrodes of the EGT, forming the transistor channel. In the next step, the substrate was annealed at  $400^\circ\text{C}$  for two hours, where the precursor-made  $\text{In}_2\text{O}_3$  inorganic semiconductor was built. Subsequently, the CSPE was printed over the transistor channel to obtain the gate dielectric. Finally, the conductive ink PEDOT:PSS was printed to form the top-gate contact between the gate electrode and the electrolyte, as proposed in [51]. For the printing steps, a Fujifilm Dimatix DMP-2831 materials inkjet-printer was used (for some experiments also the similar DMP-2850). More details on the circuit fabrication are provided in Appendix A.

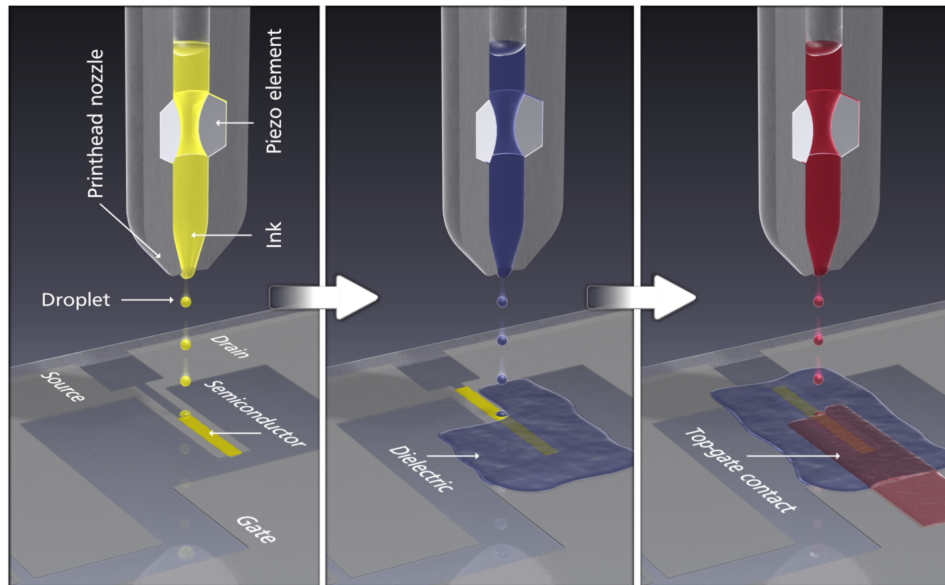
The fabricated circuits in this work were contacted using a Süss Microtech probe station, which contains low-resistance measurement needles placed at the circuit locations of interest. The probe station was connected to an Agilent 4156C parameter analyzer, to apply the supply voltages, and a Keithley 3390 arbitrary waveform generator, to generate pulsed voltage signals. Voltage signals were recorded with the Yokogawa DL6104.

For the microscopic photos, a LEICA DMLM microscope was used. For the larger circuits, multiple microscopic photos were stitched by GNU Image Manipulation Program (GIMP).

## 2.5. Exploration of New Designs and Computing Paradigms

As EGT technology was proposed very recently [25], only few prior work exist on circuit- and logic-level implementations. Most EGT-based circuits were realizations of digital components such as NAND, NOR or XOR logic gates as well as inverter-based ring-oscillators [51]. These components are already sufficient to implement any Boolean-based digital combinational circuits. However, for many applications, also sequential operations have to be performed by the deployment of memory elements. So far, no investigations were conducted how memory elements can be implemented in EGT technology. Also, the benefits and capabilities of inkjet-printing with comparison to screen- and roll-to-roll processes have not been evaluated, such as support of customizable hardware. Moreover, also from a system-level perspective, it is still an open research question how PE can be leveraged to solve classification tasks for near-sensor

## 2. Background and State of the Art



**Figure 2.7.:** *Printing steps to fabricate EGTs: **Left:** Deposition of the semiconductor ink **Middle:** Printing of the electrolyte (CSPE) **Right:** Printing of the conductive ink for top-gate contact*

computations. As it is demonstrated in this work, unconventional computing paradigms can offer several benefits, such as reduction in circuit delay, area usage, and power consumption. These aforementioned limitations are addressed in the following chapters.

## 3. Inkjet-Printed Digital Circuits

In this chapter, two fundamental digital operations are discussed. In the first section, a level-sensitive latch implemented in EGT-technology is evaluated, which is capable of storing one bit of information. The second section introduces an EGT-based low-complex lookup table, which can realize any 2-input digital logic function.

### 3.1. Inkjet-Printed SR-latch

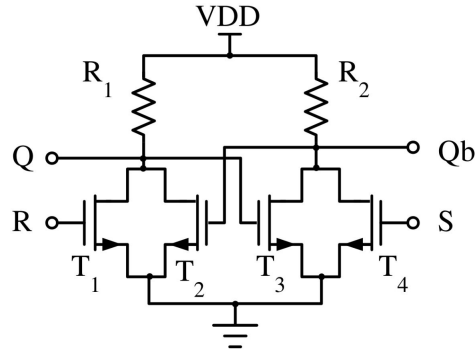
Memory cells are one of the fundamental building blocks for digital computing. In conjunction with combinational circuits, finite state machines, computer memories and sequential operations can be realized. In general, a memory cell stores one bit of binary information and by replication and aggregation of multiple cells, large-scale memory arrays can be built. Several circuit designs are deployed for the implementation of a 1-bit digital memory based on requirements such as level- or edge-sensitive operation, power consumption, performance and area usage. As low-complex designs are targeted in PE, memory designs with small number of transistors are the preferred choice. In this regard, an SR-latch is a compact circuit, which can be realized by only four transistors. To this end, the design, simulation, fabrication and measurement of an EGT-based printed latch is discussed in this section, from which more complex memories such as edge-sensitive flip flops can be derived.

#### 3.1.1. Latch Design

The latch is implemented as a Set-Reset (SR) latch realized by two cross-coupled NOR-gates, whose equivalent circuit is depicted in Figure 3.1. Due to the scarcity of p-type transistors in EGT-based technology, the pull-up network consists of resistors ( $R_1, R_2$ ). As can be obtained from the circuit diagram, the latch output  $Q/Qb$  can be changed by applying voltage signals to the S (Set) and R (Reset) port. The Boolean function of the 1-bit latch is provided in the truth table illustrated in Table 3.1. The state and output voltage are kept constant, when both input signals S and R are at logic '0'. Output transitions occur when either one of the input signals S or R are at logic '1'. Pulling both input signals high to logic '1' at the same time is not a valid operation and is usually avoided by peripheral circuitry.

#### 3.1.2. Simulation Environment

The resistor sizing ( $R_1/R_2$ ) was performed in the Cadence Virtuoso circuit simulation environment, which was extended by the printed process design kit (PPDK) [27]. The PPDK provides a behavioral DC model of the EGT. In order to enable also transient simulations, which are important for the dynamic characterization of the latch, the DC model was extended



**Figure 3.1.:** *Equivalent circuit of the SR-latch*

**Table 3.1.:** *Truth Table SR-Latch - Inputs: R,S Outputs: Q (non-inverted) and Qb (inverted)*

Reset (R)	Set (S)	$Q_{n+1}$	$Qb_{n+1}$
0	0	$Q_n$	$Qb_n$
0	1	1	0
1	0	0	1
1	1	-	-

by parasitic gate capacities, derived from the switching behavior of a printed ring oscillator [51].

### 3.1.3. Fabrication

The fabrication of the latch was according to Section 2.4. For structuring the conductive tracks of the ITO-glass substrate, eBeam lithography was deployed. The pull-up resistors were realized by meander structures obtained from the lithography step. The width and length of the EGT-semiconductor channel was set to  $575\mu\text{m}$  and  $40\mu\text{m}$ , respectively. For EGT-printing, the Dimatix DMP-2831 was used.

### 3.1.4. Measurement Setup

The electrical characteristics were measured at room temperature (295K) and with a constant humidity level at 50%. Voltage signals were generated and measured as described in Section 2.4. For circuit contacting, the Süss Microtech probe station was deployed. The measuring needles were placed directly at the conductive tracks of the fabricated circuit, their exact location along the conductive ITO tracks is however not influencing the measuring result, as long as the needles are contacted within the same ITO track<sup>1</sup>.

<sup>1</sup>Individual ITO tracks are defined as connecting paths between the circuit components such as resistors and transistors



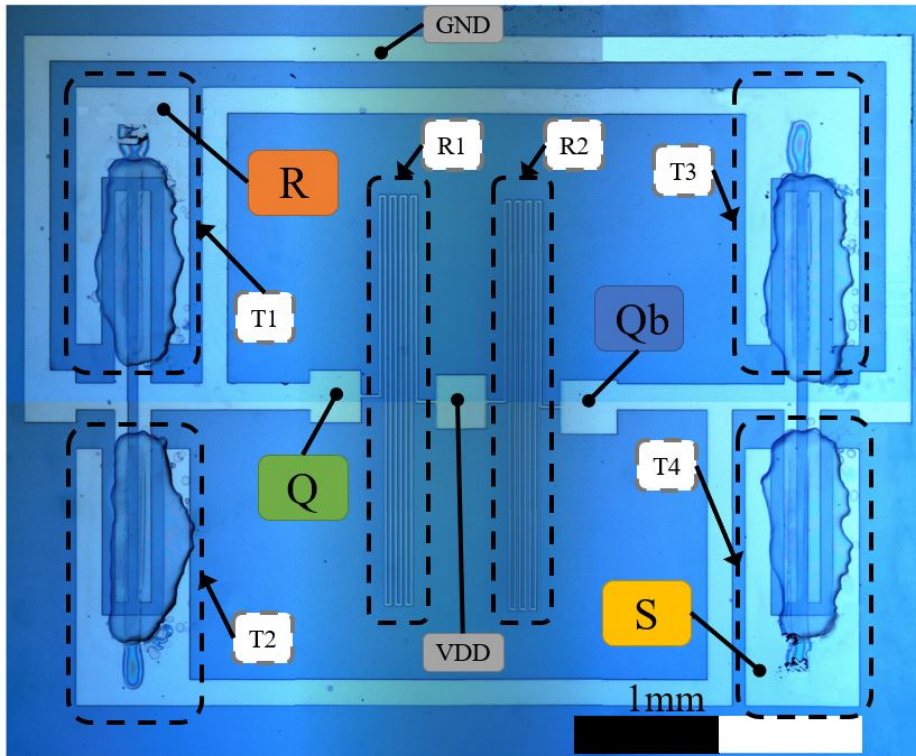


Figure 3.2.: Annotated microscopic photo of the fabricated SR-Latch. Image was digitally stitched from multiple microscope photos.

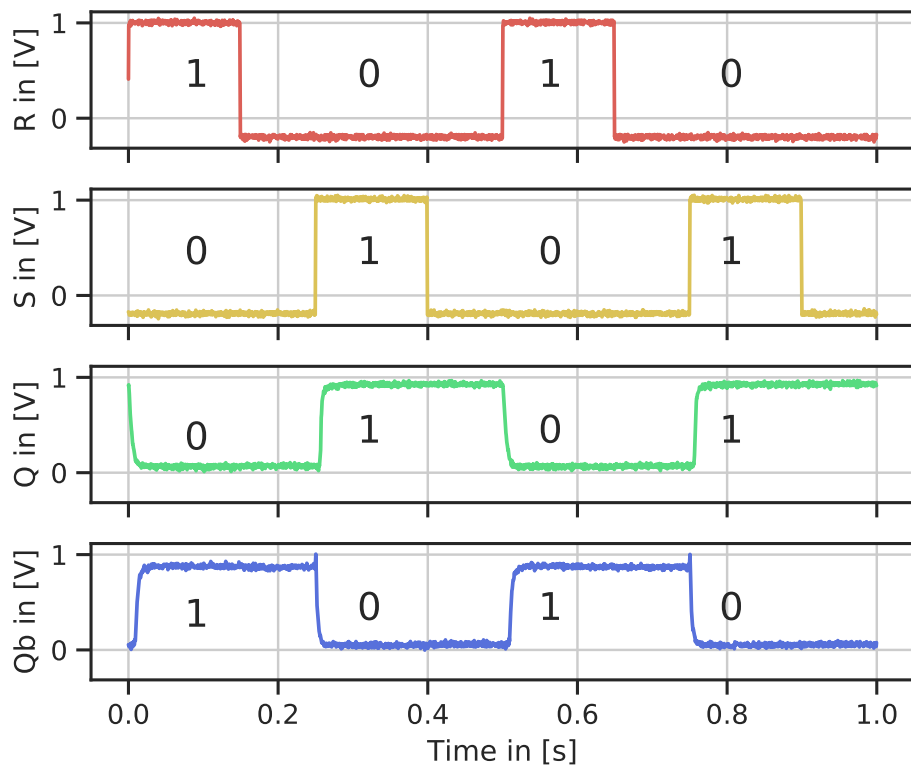


Figure 3.3.: Transient measurements of the printed SR-latch.

#### 3.1.5. Results

The microscopic photo of the inkjet-printed latch is provided in Figure 3.2. Besides the circuit components (transistors  $T_1$ - $T_4$ , resistors  $R_1, R_2$ ) also the locations are indicated, where the supply voltage ( $V_{DD}/GND$ ) and input signals ( $R, S$ ) were applied, and where the latch outputs ( $Q, Qb$ ) were measured.

First, the functionality of the fabricated circuit was tested by measuring the transient waveforms of the latch outputs in dependence on the latch input signals. For all input signal combinations the output waveform was compared to the Boolean truth table of the latch (see Table 3.1). The transient analysis is shown in Figure 3.3. As can be obtained from the measured waveforms, the output  $Q$  is pulled up to logic '1' when the set signal ( $S$ ) is logic '1', and pulled down to logic '0' when the reset signal ( $R$ ) is logic '1'. If both input signals are low, the output voltage is kept constant. In addition, the inverted output  $Qb$  is always at opposite voltage levels as  $Q$ . In summary, the printed latch was functional.

Next, the latch delay was characterized. From the transient measurements, the rise and fall time of the latch output was extracted, and the worst-case input-output delay was calculated. Despite the typically high parasitic gate capacitances of the EGTs, rise and fall times of about 3ms were obtained and an input-output delay of 4ms was measured, which leads to a theoretical operation frequency of 250Hz. Moreover, for a supply voltage of 1V, it was observed that output voltage levels range from 75mV to 850mV, which shows good rail-to-rail behavior.

The area requirement was about 7mm<sup>2</sup> with an average power consumption of about 15 $\mu$ W. The printed latch was also tested at different supply voltages and was functional down to 600mV, with reduced power consumption of 4.8 $\mu$ W but increased delay of 42ms. In terms of lifetime, only small reduction of performance was observed during the whole measurement period of about 4 weeks.

#### 3.1.6. Summary

In this chapter, an inkjet-printed SR-latch was presented, which can be used as a 1-bit memory element in sequential circuits. Due to the scarcity of p-type transistors in EGT-technology, the latch circuit differs from digital CMOS-based designs as p-type transistors are replaced by resistors. Thus, the latch consisted of only four EGTs. It was proven that the printed latch in transistor-resistor logic was functional and allows an operation frequency of about 250Hz, which is comparable to state-of-the-art organic devices. The lowest supply voltage at which the latch was functioning was only 600mV, which is exceptional compared to other printed technologies. Thus, the latch is compatible with low-power systems and a promising candidate for memory devices in future portable printed digital electronics.

## 3.2. Inkjet-Printed Lookup-Table

Lookup tables (LUT) are another commonly used building block for digital computations. Nowadays, the LUT concept experienced wide applicability especially in the context of re-configurable computing architectures for silicon-based field-programmable gate arrays (FPGA). However, also for printed electronics hardware, LUTs are a promising candidate for the efficient implementation of digital Boolean logic functions. In general, an LUT is an electronic component, which can realize any n-input digital Boolean function. Usually, a conventional LUT contains and interfaces a memory array whose stored data can be pre-calculated during an initialization phase. By deployment of additional combinational circuits, these memory cells are connected to the output of the LUT. While the combinational circuit - usually implemented by multiplexer, passing transistors or transfer gates - is a static component of the LUT, the Boolean function represented by the LUT is defined dynamically by the data stored in the digital memory.

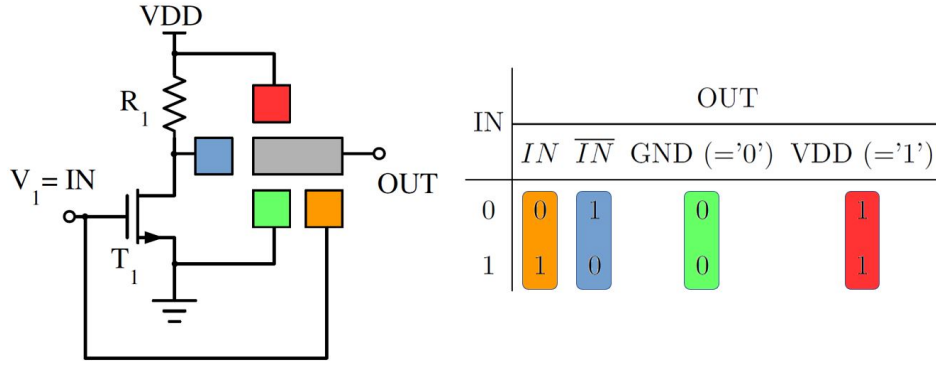
This concept can also be leveraged for digital computing architectures in PE. The LUT is in particular useful as it leads to low-complex digital printed hardware, possibilities to introduce fault tolerance against hardware failures, and lowering production time and costs of printed circuitry. However, existing CMOS-based designs cannot be directly mapped to PE due to two reasons. First, due to the scarcity of p-type transistors in EGT-technology, CMOS-based designs have to be adapted or are not feasible to be printed at all. Second, PE as an additive manufacturing process enables circuit customization and on-demand fabrication which are not an option in subtractive silicon-based technologies. For instance, the LUT memories can be replaced by hard-wiring the functionality in a customization step, by printing conductive connections at specific locations. Due to this, the resulting one-time programmable LUT (pLUT) requires only a small amount on EGTs, which improves area usage, production yield and manufacturing costs.

### 3.2.1. Proposed Printed Lookup Table Design

An n-input LUT can in total realize  $2^{(2^n)}$  different logic functions. For instance, a 1-input LUT (LUT1) is capable of implementing 4 different Boolean functions, as illustrated in Figure 3.4, which are: identity (input equals output), inverter (output opposite of input), always logic '0' and always logic '1'. An EGT-based circuit which implements this functionality is also presented in Figure 3.4. The fabrication of the one-time programmable LUT1 (PLUT1) is divided into two steps. First, the EGTs, resistors and conductive tracks are fabricated to implement an inverter using the process described in Section 2.4. In the second step, a conductive ink is deployed and printed as a connection between the output pad and one of the other four pads (Figure 3.4), which defines the one-time-programmed functionality of the PLUT1. The output function of the PLUT1 can also be denoted as:

$$OUT = f(IN) = \begin{cases} 0, & f := 0 \\ 1, & f := 1 \\ IN, & f := id \\ \overline{IN}, & f := inv \end{cases} \quad (3.1)$$

### 3. Inkjet-Printed Digital Circuits



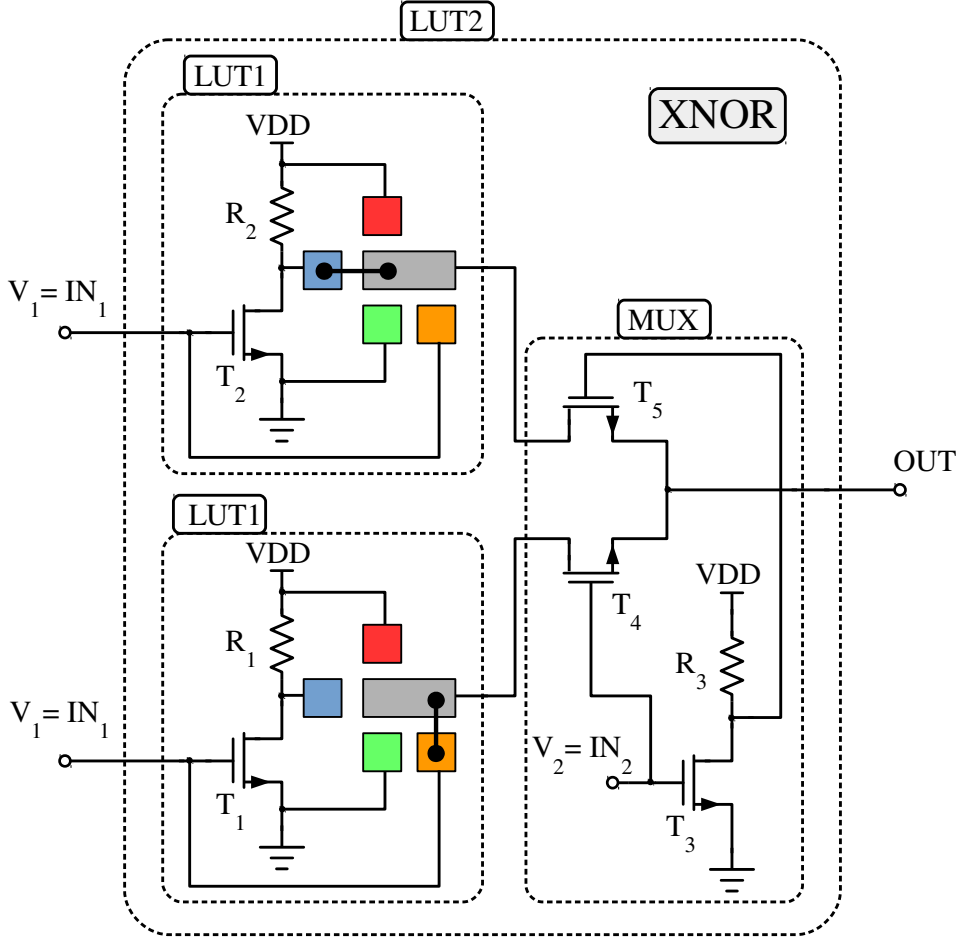
**Figure 3.4.:** Schematic and truth table of the proposed PLUT1. By printing low-resistance connections between the output pad (grey pad), and one of the colored pads, the Boolean function of the PLUT1 is defined. E.g., by connecting the output pad and the blue pad, the circuit behaves as an inverter. Conversely, connecting the output pad with the orange pad leads to the identity function.

, where  $IN$  is the PLUT1 input signal and  $OUT = f(IN)$  the PLUT1 output signal. As mentioned before, 4 different Boolean functions can be realized by the PLUT1. They are abbreviated in the following by: '0', '1', 'id' and 'inv'.

The PLUT1 can be extended to a programmable 2-input LUT (PLUT2), by replication of the PLUT1 and adding a passing-transistor-based 2-input multiplexer (see Figure 3.5). By configuration and hardwiring the LUTs, any possible output function  $OUT(IN_1, IN_2)$  is realized. The output function of the PLUT2 is denoted as:

$$OUT(IN_1, IN_2) = (f_1(IN_1) \wedge IN_2) \vee (f_2(IN_1) \wedge \overline{IN_2}), \quad (3.2)$$

where  $IN_2$  is the multiplexer select signal and  $IN_1$  the PLUT1 input signal, and  $f_1(\cdot), f_2(\cdot)$  are two different PLUT1 functions as defined in Equation (3.1). In total, a 2-input LUT can be programmed to implement  $2^{(2^2)} = 16$  different 2-input Boolean functions, by choosing instantiations for  $f_1(\cdot)$  and  $f_2(\cdot)$  according to Equation (3.1). As an example, in Equation (3.3), 10 common Boolean functions are realized, by choosing different 1-input functions for  $f_1(\cdot)$  and  $f_2(\cdot)$ . In general, any kind of logic can be implemented, such as a logic OR, AND, NOR, NAND, XOR or XNOR operation.



**Figure 3.5.:** Schematic of a printed 2-input lookup table (PLUT2) programmed as an XNOR operation

$$\text{OUT}(IN_1, IN_2) = \begin{cases} 0, & \text{if } f_1 = 0 \wedge f_2 = 0 \\ 1, & \text{if } f_1 = 1 \wedge f_2 = 1 \\ IN_1, & \text{if } f_1 = id \wedge f_2 = id \\ \overline{IN_1}, & \text{if } f_1 = inv \wedge f_2 = inv \\ IN_2, & \text{if } f_1 = 1 \wedge f_2 = 0 \\ \overline{IN_2}, & \text{if } f_1 = 0 \wedge f_2 = 1 \\ IN_1 \vee IN_2, & \text{if } f_1 = 1 \wedge f_2 = id \text{ (OR)} \\ IN_1 \wedge IN_2, & \text{if } f_1 = id \wedge f_2 = 0 \text{ (AND)} \\ \overline{IN_1 \vee IN_2}, & \text{if } f_1 = 0 \wedge f_2 = inv \text{ (NOR)} \\ \overline{IN_1 \wedge IN_2}, & \text{if } f_1 = inv \wedge f_2 = 1 \text{ (NAND)} \\ IN_1 \oplus IN_2, & \text{if } f_1 = inv \wedge f_2 = id \text{ (XOR)} \\ IN_1 \Leftrightarrow IN_2, & \text{if } f_1 = id \wedge f_2 = inv \text{ (XNOR)} \end{cases} \quad (3.3)$$

It is important to mention, that the PLUT2 can also be realized by one PLUT1 instead of two [29], however this additional redundancy can be leveraged to enable re-routing and fault tolerance capabilities for yield enhancements.

### 3. Inkjet-Printed Digital Circuits

With only 5 EGTs, the proposed PLUT2 requires less EGTs compared to state-of-the-art designs, which are the logic gate (LG)-based LUT (10 EGTs) and the pass transistor (PT)-based LUT (8 EGTs) [52]. Another state-of-the-art LUT, which is the transmission-gate-based LUT, cannot be realized in EGT technology due to the absence of p-type transistors.

A comparison of the different PLUT2 implementations is provided in Table 3.2. The proposed PLUT2 improves all pertinent parameters such as worst-case delay, area usage and power consumption, however at the expense of reduced rail-to-rail behavior. This is due to the fact, that the logic-1 level of the PLUT2 output voltage is not reaching the supply voltage, which was 1V in this experiment. However, this could be compensated by an additional (inverter-based) output buffer.

#### 3.2.2. Hardware Prototype

As a proof of concept, the PLUT2 was fabricated and characterized according to Section 2.4. For the hardware prototypes, laser ablation on the ITO substrate was deployed to obtain the conductive tracks, which are meander-shaped resistors and EGT electrodes. The EGT semiconductor channel width/length was set to 200 $\mu\text{m}$ /80 $\mu\text{m}$ . The pullup-resistors were set to 100k $\Omega$ . The supply voltage was set to 1V and measurements were performed at constant humidity level of 70%.

The microscopic photo of the printed PLUT2 is depicted in Figure 3.6. The additionally conductive connections printed to the PLUT1s, are marked by a red box. These connections define the functionality of the circuit (see Figure 3.4). For example, the circuit shown in Figure 3.6 implements an XNOR function, which can also be derived from the measured waveforms, provided in Figure 3.7. Thus the connections were printed as illustrated in Figure 3.5.

By changing the location of the printed connections at the PLUT1 output pads, other functionalities can be realized. E.g. the circuit in Figure 3.8 represents an XOR gate, while Figure 3.10 implements an AND gate, as can be obtained from the measured waveforms in Figure 3.9 and Figure 3.11, respectively.

From the waveforms it becomes obvious, that not the full output voltage swing is provided by the printed circuits. This is due to the voltage drop at the passing transistors when a logic '1' has to be propagated. Nevertheless, the logic '1' signals are clearly differentiable to the logic '0' signals and can be further amplified by an additional buffer element at the output, such as two cascaded inverters.

The average power consumption of the hardware prototypes was 25.12 $\mu\text{W}$  with a worst case

**Table 3.2.:** Comparison of LUT2 implementations based on EGT technology [29]

	Delay	Area	Power (average)	Logic-1 output voltage
LG-based	13.3ms	120mm <sup>2</sup>	193 $\mu\text{W}$	1V
PT-based	2.8ms	28mm <sup>2</sup>	30 $\mu\text{W}$	0.6V
pLUT2	2.7ms	17.4mm <sup>2</sup>	25 $\mu\text{W}$	0.8V

delay of 73.28ms and area usage of 60mm<sup>2</sup>. These numbers differ from the simulation-based evaluation presented in Table 3.2, as for the simulations an optimal design was considered, e.g. a circuit without measurement pads as used for the hardware prototypes.

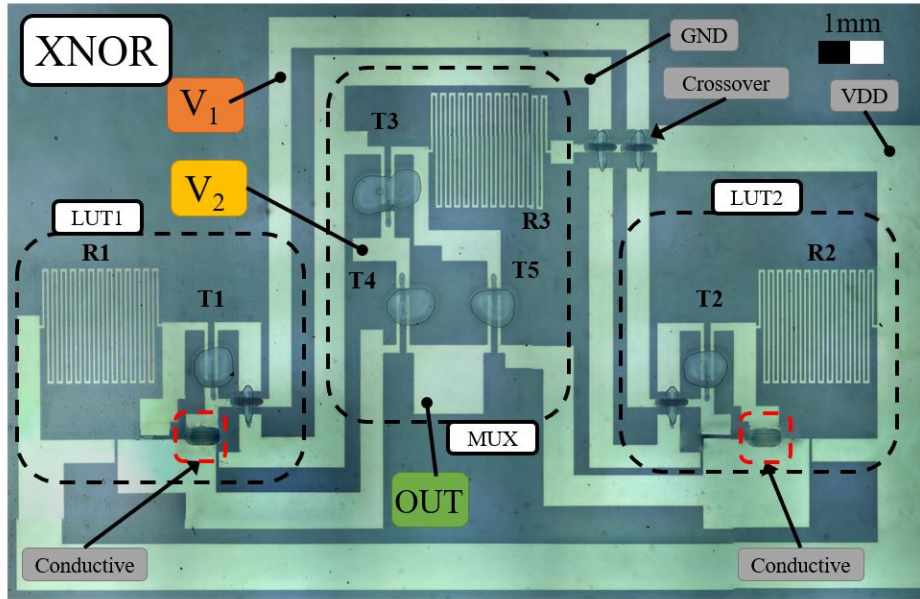


Figure 3.6.: Digital post-processed image from multiple microscope photos of the printed LUT2 configured as an XNOR

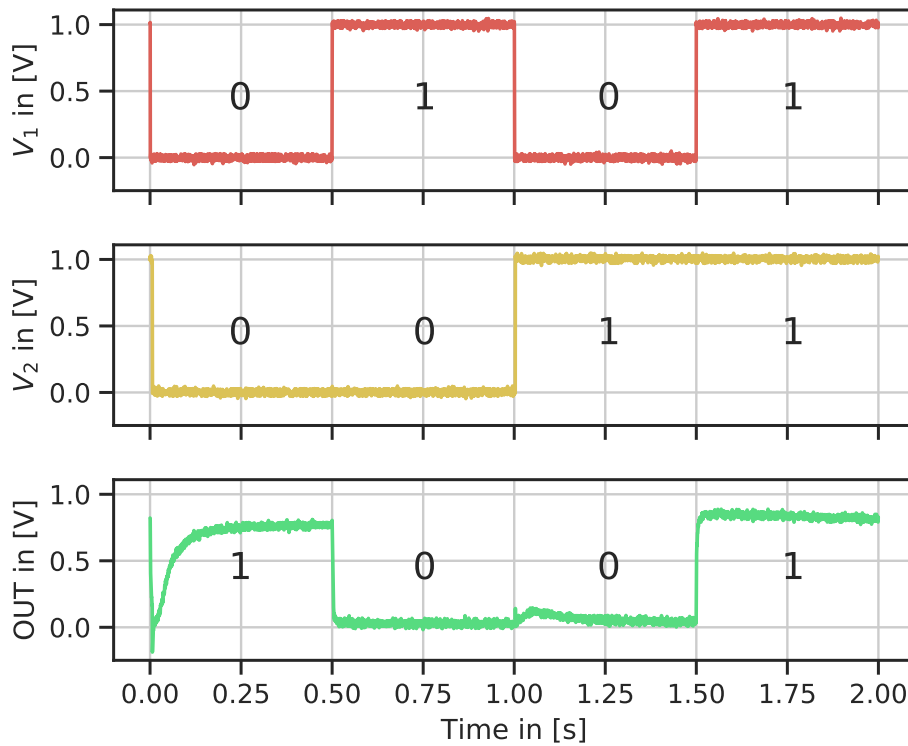


Figure 3.7.: Measured waveforms of the printed LUT2 configured as an XNOR

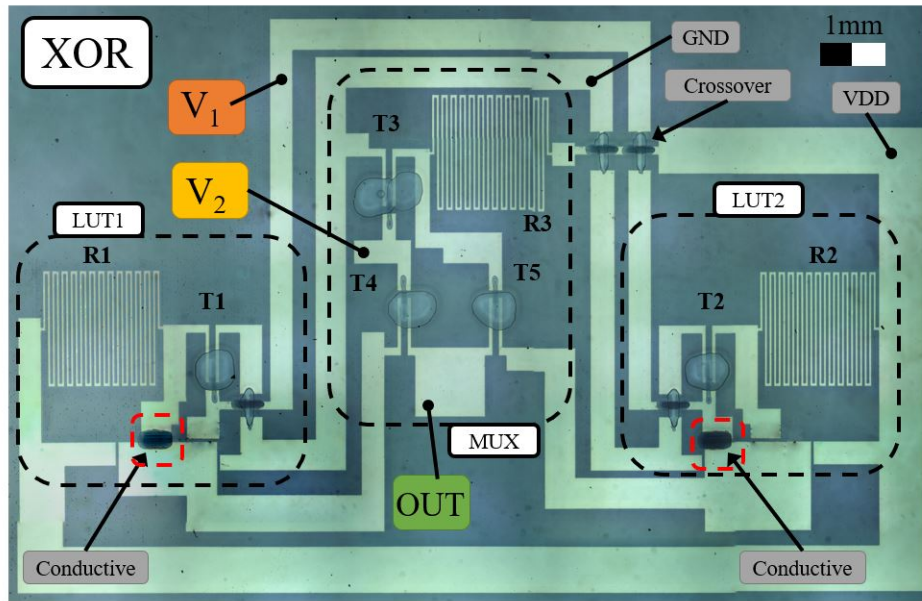


Figure 3.8.: Digital post-processed image from multiple microscope photos of the printed LUT2 configured as an XOR

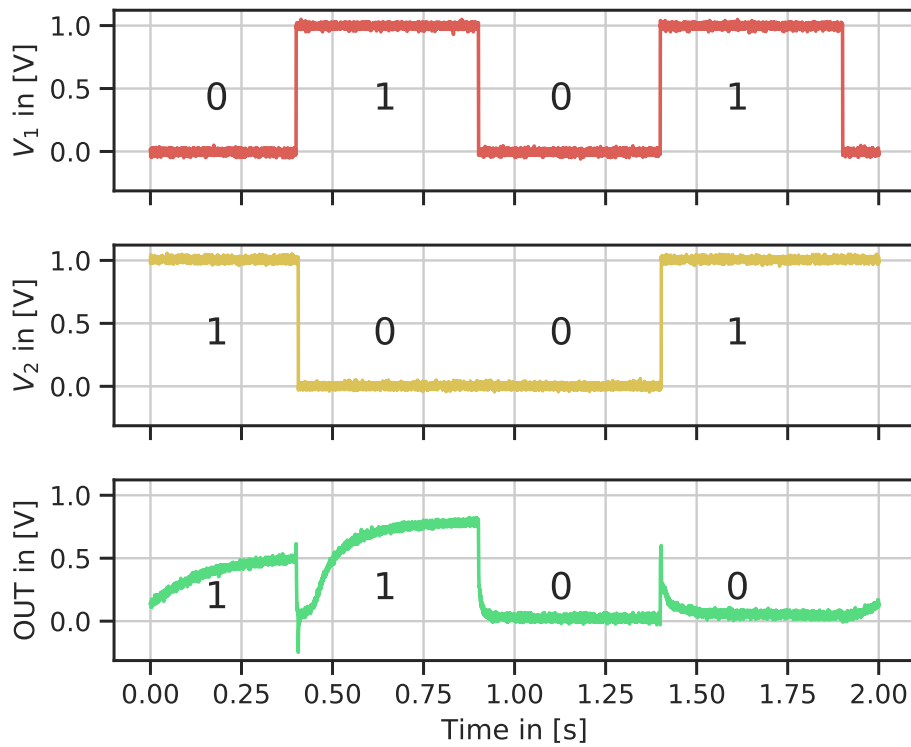
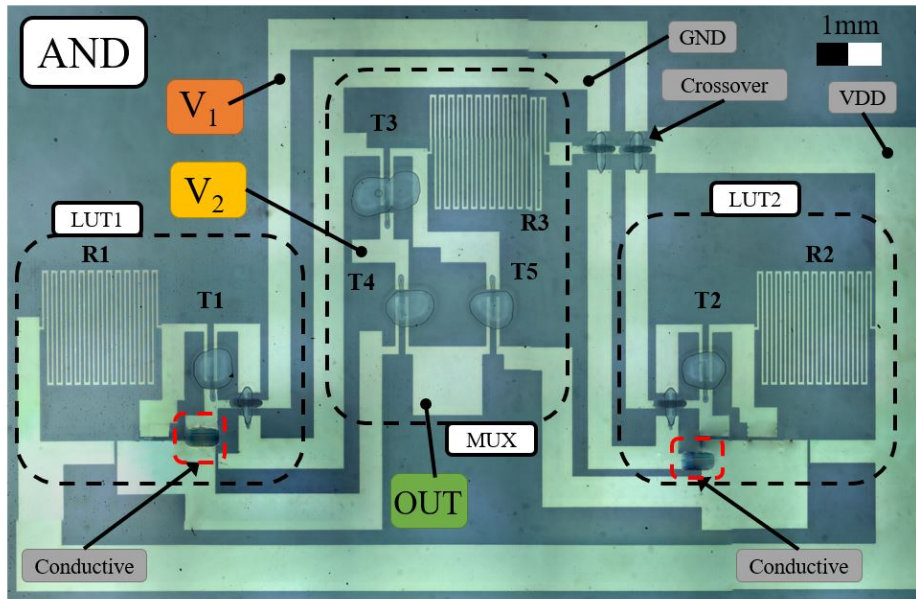


Figure 3.9.: Measured waveforms of the printed LUT2 configured as an XOR

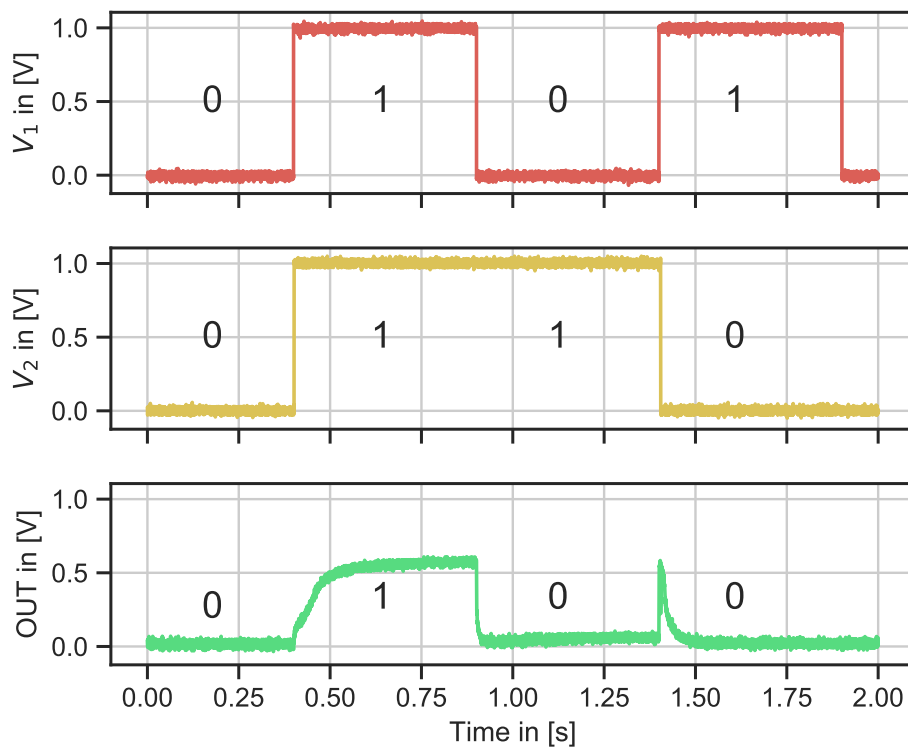
### 3.2.3. Summary

In this section, the design, simulation, fabrication and characterization of a printed 2-input LUT was presented. The proposed design has lower complexity than existing LUT designs [52],





**Figure 3.10.:** Digital post-processed image from multiple microscope photos of the printed LUT2 configured as an AND



**Figure 3.11.:** Measured waveforms of the printed LUT2 configured as an AND

and consequently significantly better area usage and power consumption. As less EGTs are required for the circuit implementation, the production yield is also enhanced. Similarly, as a LUT2 is constructed from a LUT1, also an LUT3 can be built by adding another level of passing-transistor-based multiplexers. This can then be extended to LUT4 etc. However, in

### 3. Inkjet-Printed Digital Circuits

contrast to transfer gates, logic '1' signals are degraded at each passing transistor and not a full voltage swing is obtained. As a countermeasure, the degrading signal can be restored by adding buffer elements (e.g., inverter-based) at the output of the LUT2 or LUT3. Thus, also large-scale integration of the printed LUT is enabled and any combinational circuit can be realized.

Another advantage of the proposed design is the split-manufacturing capability. For instance, in order to improve the low production throughput and overall production costs of inkjet-printing technology, high-volume fabrication techniques such as roll-to-roll processes can be deployed to print all the standard circuit components, and then in a second step, the printed lookup tables are one-time programmed by the end user by inkjet-printing the connections at the output pad of the LUT. This enables on-demand and customized designs.

Finally, the proposed LUT design can be used to mask hardware failures by rerouting and bypassing defective components, which further increases the chip yield. Defective parts can be identified using digital testing methods and then fault tolerance is introduced as deployed in re-configurable computing systems [53].

## 4. Inkjet-Printed Neuromorphic Architectures

As it was presented in Section 3.2, the complexity of digital designs in PE can be substantially reduced by using the customization capabilities of the printing process. However, due to the large feature sizes in PE, which are in the micrometer range - compared to nanometer in silicon technology - functional density of printed hardware is still very low. Due to this, Boolean digital logic designs lead to high area usage, high power consumption and low performance [50], which limits the applicability to several application domains. Moreover, process variations in PE are significant and can induce performance fluctuations or even hardware failures. As a result, these limitations favor the use of unconventional computing architectures in PE, e.g. for near-sensor processing applications, which have stringent requirements on area, performance and reliable operation [33]. In this chapter, printed neuromorphic hardware is proposed as a complexity reduction technique. In Section 4.1, the hardware implementation of a feed-forward artificial neural network is proposed, which introduces EGT-based circuits for all fundamental building blocks for realization of neural algorithms. In Section 4.2, another type of neuromorphic hardware is explored, which is based on stochastic computing, to time-multiplex multi-valued neural network inputs and weights to simplify the implementation of arithmetic operators.

### 4.1. Inkjet-Printed Analog Artificial Neural Network

Brain-inspired aka neuromorphic computing can overcome shortcomings of conventional hardware by its analog implementation of neural algorithms. In its analog continuous-state representation, a neuromorphic computing system (NCS) can directly process sensory data without converting it into digital multi-bit representations, which would otherwise require expensive analog/digital converters (ADCs). Moreover, analog signal processing increases the functional densities, which improves the hardware footprint of printed NCS, as well as the power consumption. For instance, a digital designed printed 3-input neuron with 4-bit precision requires 357 printed transistors, while an analog implementation providing the same functionality requires only four transistors (see Section 4.1.4). Also performance-wise enhancements emerge due to the parallel computing capabilities of NCS in contrast to sequential circuits in digital computing architectures.

In addition, as NCS hardware is tailored to a specific target application, similar to the proposed LUT presented in Section 3.2, one-time programming can be achieved by customization of inkjet-printed neural network weights, which further improves performance, power consumption and area requirements. Finally, in order to increase the low chip yield in PE for ensuring reliable operation, a printed NCS can be trained to be intrinsically fault tolerant [54, 55].

### 4.1.1. Artificial Neural Networks (ANN)

Popular realizations of NCS are either based on spiking neural networks (SNN) [56] or feed-forward/artificial neural networks (ANN) [57]. SNNs are brain-inspired neural networks, where information-flow is achieved by propagation of voltage pulses, or spikes. Due to the event-driven computation, SNNs yield very high energy efficiencies. On the other side, ANNs are more akin to the popular neural networks developed for GPU- or CPU-platforms. Here, signals are represented by real-valued quantities and processing is performed according to the McCulloch-Pitts neuron model [57]. The ANN can approximate any functionality and smooth function by using two basic hardware components: non-linear activation function and multiply-accumulate (MAC) also called weight aggregation or weighted sum [58]. Due to its simplicity, training in ANNs is achieved by the well-known least-mean-square learning rule and are thus commonly used in hardware implementations. The training algorithm in ANNs converges very fast to a solution due to the back-propagation-based learning algorithm [59] and thus ANNs could demonstrate broad applicability [57]. Due to the unique customization feature of inkjet-printed technology, new ANN topologies can be explored such as sparse ANN interconnections and low-device-count MAC operations. This yields in gains in ANN size and speed, as well as simplifications in ANN learning and inference.

Examples of printed NCS components have been recently presented. In [20] a MAC engine on flexible substrates with a time domain encoded implementation was introduced. The authors of [60] presented an organic-based crossbar-architecture, which also realizes the MAC operation. Also neuron activation functions were presented besides organic MAC operations. In [61], a low-complexity design for activation function circuits based on organic p-type transistors for implementation of an ANN was reported.

Although many of the aforementioned contributions realize parts of a printed NCS, they either do not provide all fundamental building blocks [60], or are not based on printable materials [61]. In this section, this gap is bridged by providing circuit designs in PE for all ANN building blocks, including comprehensive ANN weight representation and non-linear activation functions. Hardware prototypes are provided as a proof of concept, as well as a learning algorithm tailored to the properties of PE technology. Parts of the results are identical with [30].

### 4.1.2. Printed Programmable Neuron

As discussed before, two fundamental building blocks are required to implement ANNs: MAC operations and non-linear activation functions. While circuits for these operations can be easily derived for digital-based Boolean logic, analog counterparts are non-trivial and require novel designs, especially under the technology-dependent restrictions of PE. The analog signal processing allows for continuous inputs, outputs and weights representation in the ANN, which would otherwise require many hardware resources in digital implementations. Besides digital implementations, also existing analog solutions such as operational amplifiers are currently beyond the capabilities of PE due to their high transistor count.

Another limitation of PE is related to the low number of available printable functional materials. Although, the requirements for transistor printing are already fulfilled, the fabrication of inkjet-printed memristors remains still an open research question. Therefore, printed resistor-

based crossbars are proposed, however without sacrificing the customization feature of crossbars due to the additive fabrication technology in PE. In order to map a learned set of ANN weights to the ANN hardware, inkjet-printed conductive ink is deposited to the crossbar interconnects, with different lateral geometries and layer thicknesses to realize a wide range of resistors, and thus ANN weights.

### Analog MAC Operation

The MAC operation is a standard operation for performing ANN inference. At each ANN node, the MAC operation - as part of a single neuron - adds up the inputs, scaled and weighted by the ANN weights  $w_i$ . The output  $a$  of the MAC operation can be computed by [58]:

$$a = \mathbf{x}^T \mathbf{w} = \sum_i w_i x_i + w_b x_b = \sum_i w_i x_i + b, \quad (4.1)$$

where  $x_i$  are the neuron input signals and  $x_b$  is a constant input, thus  $w_b x_b$  is denoted as the constant neuron bias  $b$ .

For ANNs implemented in software and running on CPUs, this operation is usually performed sequentially in the floating point unit, or in parallel on Digital Signal Processors (DSP). In contrast, the implementation of MACs for printed neuromorphic hardware is much different. First of all, the inputs of a neuron  $x_i$  are encoded as voltage signals  $V_i$ , which are applied to a crossbar architecture with printed resistors  $R_i$  at the crossbar interconnects (see Figure 4.1a). According to Ohm's law, the voltages across the resistors ( $V_i - V_x$ ) are converted into currents, and these currents are summed up following Kirchhoff's rule. The output voltage  $V_x$  of the crossbar can be computed similar to a Y-circuit (i.e., a circuit where one port of each resistor is connected together). As only resistors are contained, the crossbar resembles a linear circuit and the output voltage  $V_x$  can be computed analytically (a more detailed mathematical derivative of the MAC formulas is provided in the Appendix B).

The relationship between the neuron input voltages  $x_i = V_x$  and the neuron weights  $w_i$ , as well as the constant bias voltage  $b = V_{bias} \cdot w_b$ , is as follows:

$$V_x = \sum_i V_i w_i + V_{bias} w_b. \quad (4.2)$$

For simplicity, in the following the resistors  $R_i$  are denoted with the conductances  $g_i = \frac{1}{R_i}$ , or  $g_b = \frac{1}{R_b}$  and  $g_d = \frac{1}{R_d}$  respectively.

Thus, the synaptic weights can be abbreviated by:

$$w_i = \frac{g_i}{\left(\sum_j g_j\right) + g_b + g_d}, \quad (4.3)$$

and for the bias weight:

$$w_b = \frac{g_b}{\left(\sum_j g_j\right) + g_b + g_d}, \quad (4.4)$$

#### 4. Inkjet-Printed Neuromorphic Architectures

and the decoupling weight:

$$w_d = \frac{g_d}{\left(\sum_j g_j\right) + g_b + g_d}. \quad (4.5)$$

The crossbar output  $V_x$  behaves like a MAC operation (Appendix B):

$$a = V_x = \sum_i w_i V_i + w_b V_{bias} = \sum_i w_i x_i + b. \quad (4.6)$$

Moreover, from Equation (4.3) and Equation (4.4) it is obvious that  $w_i$  and  $w_b$  are lower- and upper-bounded:  $w_i, w_b \in [0, 1]$ . A reason for the lower bound is the fact that resistors are physically only positive. On the other side, the upper bound can be explained that in a passive resistor network applied voltages cannot be increased, only reduced due to power dissipation.

Another constraint can be obtained by summing up all  $w_i$ ,  $w_b$  and  $w_d$  which results in the second constraint:

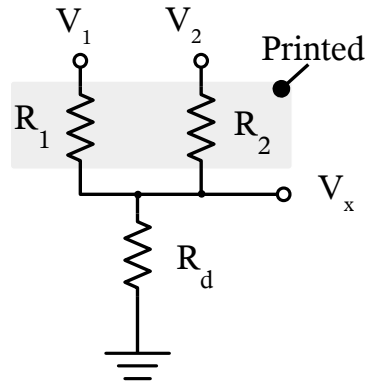
$$\sum_i w_i + w_b + w_d = 1. \quad (4.7)$$

It is important to note that the above Equation (4.7) with coupled weights can be decoupled by proper adjustment of the conductor/resistor  $g_d/R_d$  (also called  $R_{base}$ , see [61]). The decoupling resistor  $R_d$  is added to the resistive crossbar similar to  $R_i$  and  $R_b$ , however a constant voltage of  $0V$  is applied to it (i.e.  $V_d = 0V$ ), see Figure 4.6. As a result, the decoupling resistor can be used as a placeholder to adjust the weight formula (Equation (4.3) and (4.4)) without biasing the MAC output  $V_x$  (Equation (4.6)).

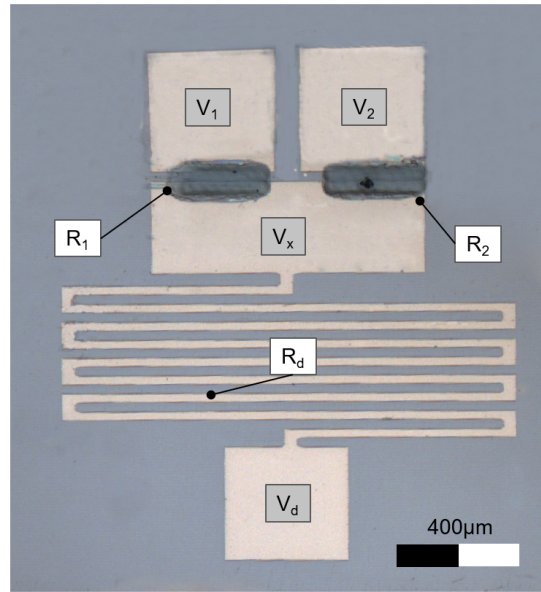
In Figure 4.1a, a circuit for a 2-input MAC operation is depicted. The weights  $w_i$  of the MAC operation depend on the resistance values of  $R_1$  and  $R_2$ , which are inkjet-printed and customized according to the ANN training step. Figure 4.1b shows the layout of the hardware prototype of the printed crossbar. The passive conductive structures, including the meander resistor ( $R_d$ ) were obtained from laser-ablated ITO-sputtered glass substrates (Section 2.4). The resistors  $R_1$  and  $R_2$  were printed using PEDOT:PSS conductive ink. This allows for customization of the neuron according to the pre-trained weights vectors.

The transient measured input and output waveforms are depicted in Figure 4.1c. Both  $V_1$  and  $V_2$  are pulsed between  $-1V$  and  $1V$ , with a pulse width of  $10ms$  and  $5ms$ , respectively. The output pulse  $V_x$  is obtained in dependence of the input pulses. The circuit response behaves as expected, showing only close to  $0V$  signals when the inputs are complementary to each other. The output  $V_x$  is pulled up or down to  $0.5V$  or  $-0.5V$  when both signals are at  $1V$  or  $-1V$ , respectively. The resulting weights were:  $w_1 = w_2 \approx 0.25$  and  $w_d \approx 0.5$ . So both coefficients of the MAC operation are set to  $0.25$ , and the correct summation and multiplication result  $V_x = w_1 \cdot V_1 + w_2 \cdot V_2 = 0.25 \cdot V_1 + 0.25 \cdot V_2$  is obtained from the measured waveform. Figure 4.1c also shows the simulated waveform extracted from the circuit simulator. The measured signal is approaching the simulation result, and confirms the correct implementation of the MAC circuit.

The choice of design parameters and the measurement results are depicted in Table 4.1.

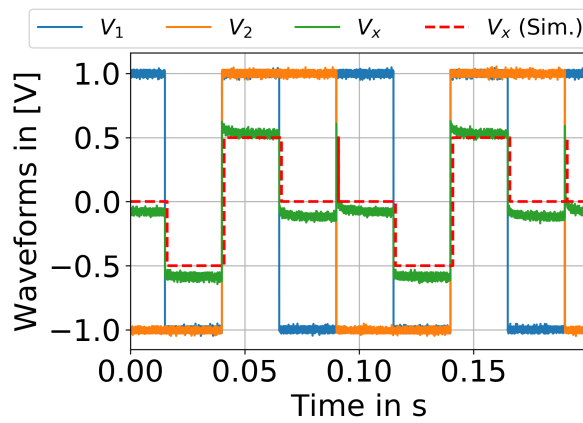


(a) Schematic



(b) Microscopic Photo

Multiply-Accumulate (MAC)



(c) Measured Waveforms

**Figure 4.1.:** (a) shows the schematic of the 2-input crossbar ( $V_1, V_2$ ) implementing the MAC operation. (b) depicts the digitally post-processed microscopic photo of the fabricated MAC circuit. (c) illustrates the waveforms from the simulation and circuit measurements.

#### 4. Inkjet-Printed Neuromorphic Architectures

**Table 4.1.:** Design parameters and measurement results of the hardware prototypes. Resistance values  $R_i, R_d$  are indicated, also the supply voltages  $VDD/VSS$  and the printed transistor channel geometries  $T_i = \text{width}/\text{length} = W/L$

Components:		MAC	inv	ptanh
Design Parameters		$R_1 = 100\text{k}\Omega$	$R_1/R_2 = 160\Omega/80\Omega$	$R_1/R_2 = 180\text{k}\Omega/80\text{k}\Omega$
		$R_2 = 100\text{k}\Omega$	$R_3/R_4 = 25\text{k}\Omega/15\text{k}\Omega$	$T_1 = 100\mu\text{m}/80\mu\text{m}$
		$R_d = 50\text{k}\Omega$	$R_5 = 80\text{k}\Omega; T_1 = 500\mu\text{m}/40\mu\text{m}$	$T_2 = 500\mu\text{m}/40\mu\text{m}$
			$VDD/VSS = 1\text{V}/-2.2\text{V}$	$VDD/VSS = 1\text{V}/-1\text{V}$
Results	Delay	1ms	4ms	4.5ms
	Power	31.3 $\mu$ W	30mW	44.8 $\mu$ W
	Area	2.6mm <sup>2</sup>	20.3mm <sup>2</sup>	25.6mm <sup>2</sup>

#### pPLU Activation Function

Activation functions are deployed in ANNs to introduce non-linear behavior to the neuron computation [58]. Due to the linear behavior of resistor-based circuits, additional components are required to obtain non-linear behavior. As the transfer functions of transistors are non-linear, they are a suitable candidate for implementing ANN activation functions [61]. The proposed ANN activation function is illustrated in Figure 4.3a. As its transfer function is a piece-wise linear unit, the term "printed piece-wise linear unit" (pPLU) is used in the following. The design of the pPLU consists of an EGT with two resistances at the gate to form a voltage-divider. The transistor is turned on for positive voltages across the voltage dividers, and off for negative voltages, similar to a diode. In order to obtain different gate voltages by changing the polarity of the voltage divider, the resistors can be sized to fulfill:  $R_L \gg R_H$ . Dependent on the polarity of the input voltage  $V_x$  of the activation function, the transistor is either in on- or off-state, and either high positive transistor drain currents or small negative drain currents flow through the pull-down resistor  $R_{out}$ . The non-linear fluctuations of the transistor drain current are converted into non-linear output voltages  $V_{out}$  of the neuron.

As the pPLU is directly connected to the MAC circuit, the resistance  $R_{out}$  must be chosen much higher than the crossbar resistors ( $R_{out} \gg R_i, R_b, R_d$ ), e.g. by a factor of 10, otherwise the resistor-to-weight calculation is biased (see Equation (4.3)-(4.5)) [61]. Similarly, the resistor  $R_L$  in the voltage divider has to be higher than the On-resistance of the EGT ( $R_L \gg R_{ON}$ ), to ensure that the transistor is not shortcut by the voltage divider.

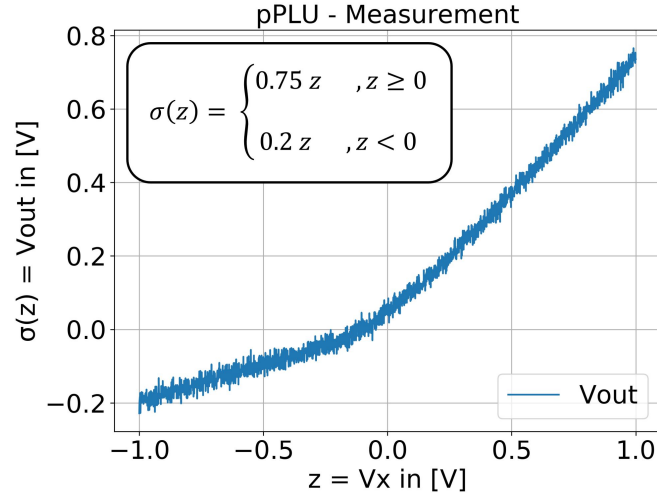
The measured waveforms of the hardware prototype are depicted in Figure 4.2, and as expected, the piece-wise behavior of the pPLU is obtained.

#### Two-input Neuron Design and Hardware Prototype

Based on the proposed MAC and pPLU circuit, a two-input neuron was designed as depicted in Figure 4.3a. The first part of the neuron circuit contains a crossbar-based MAC engine, which processes the input voltages  $V_i$ . The input signals are accumulated according to Equation (4.6) and the output  $V_x$  is applied to the pPLU activation function. The pPLU output  $V_{out}$  is then applied to the successive neuron in the next ANN layer (not part of the hardware prototype).

The fabrication of the hardware prototype was according to Section 2.4. The resistors  $R_1, R_2$





**Figure 4.2.:** Measured waveforms of the printed piece-wise linear unit (pPLU)

**Table 4.2.:** Design parameters for 2-input neuron fabrication

$R_1$	$R_2$	$R_d$	$R_{out}$	$R_L$	$R_H$	W/L of T1
$1k\Omega$	$1k\Omega$	$30k\Omega$	$300k\Omega$	$500k\Omega$	$10k\Omega$	$200\mu m/80\mu m$

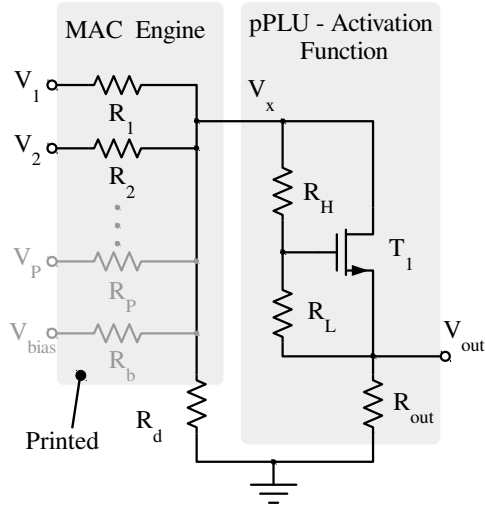
were inkjet-printed using PEDOT:PSS conductive ink. The circuit components were sized according to Table 4.2. A microscopic photo is provided in Figure 4.3b. The functionality of the printed neuron was tested by electrical characterization as described in Section 2.4. Transient measurements were performed by application of input voltage pulses  $V_1, V_2$  to the MAC circuit. The crossbar resistors were sized equally (Table 4.2), thus the resulting weights are:  $w_1 \approx w_2 \approx 0.49$  and  $w_d \approx 0.02$  (see Equation (4.3)). As can be obtained from Figure 4.3c, the input voltages are pulsed between  $-1V$  and  $1V$ . When both inputs are at  $1V$ , the output voltage  $V_{out}$  is pulled up to about  $700mV$ . On the other side, when both inputs are at  $-1V$ ,  $V_{out}$  is pulled down to about  $-300mV$ . This validates the correct functionality of the non-linear activation function, which depresses input voltages in the negative voltage range (see Figure 4.2). For complementary inputs (e.g.  $V_1 = -1V$  and  $V_2 = 1V$  or vice versa),  $V_{out}$  is close to  $0V$ , as both ANN weights  $w_1$  and  $w_2$  were chosen to be equal, which validates the correct functioning of the MAC circuit. In summary, the fabricated 2-input neuron was operating as expected and close to the simulation results (see Figure 4.3c).

The latency of the neuron was about  $1.6ms$ , which leads to a maximum operating frequency of  $625Hz$ . The area requirement was  $54mm^2$ . The average power consumption of the 2-input neuron was about  $14.8\mu W$ . The input voltage swing could be lowered to  $-0.7V/0.7V$ , where the neuron was still functioning and the output signals could still be differentiated.

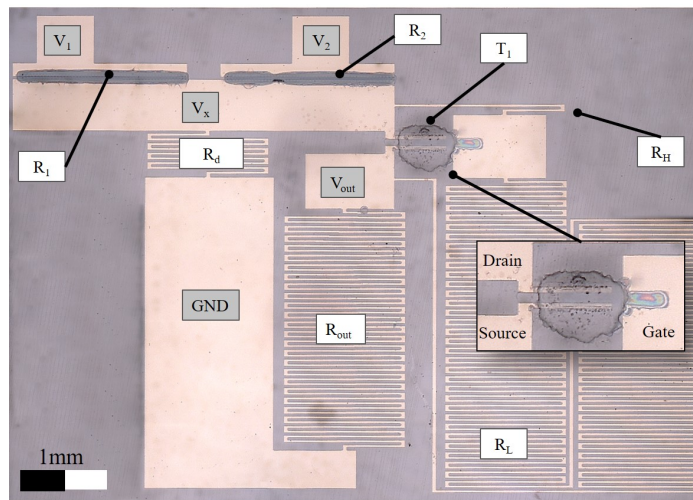
### 4.1.3. Improvements on the Proposed ANN design

Although the functionality of the proposed printed neuron presented in Section 4.1.2 was validated, there exist several limitations. One limitation is the signal degradation across ANN layers. Due to the pPLU, at each ANN layer, the output signals are smaller than the input signals (see Figure 4.2), thus, cascading of printed ANN layers leads to deterioration of electrical

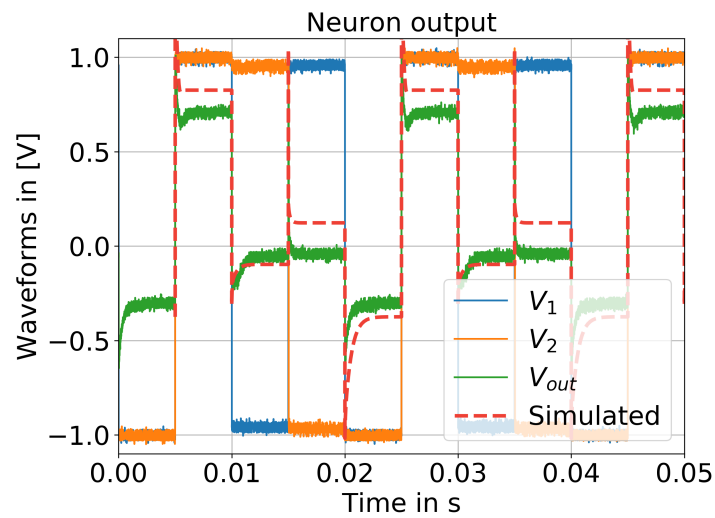
#### 4. Inkjet-Printed Neuromorphic Architectures



(a) Schematic



(b) Microscopic Photo



(c) Measured Waveforms

**Figure 4.3.:** (a) shows the schematic of the printed neuron (b) depicts the digital post-processed image from multiple microscope photos of a the 2-input neuron hardware prototype. (c) contains the measured transfer function of the circuit.

signals at the ANN output which cannot be distinguished from noise anymore. Moreover, the ANN voltage signals at the ANN output layer have to be distinguishable among themselves to obtain the correct classification outcome, which are represented by a discrete set of voltage levels. This requires a minimum sensing resolutions of the output signals, which has to be considered during ANN training. Finally, with the proposed crossbar-based MAC operation, no negative weights can be realized, which limits the applicability to many real-world classification problems.

### Negative weights

The MAC operation in a single printed neuron can perform vector operations between a variable input signal vector  $\mathbf{x}$  and a constant weight vector  $\mathbf{w}$ . However, due to the implementation by a resistor crossbar, negative multiplications between inputs  $x_i$  and weights  $w_i$  (e.g.  $w_i < 0$ ) are not possible, as the resistances/conductances are physically only positive and the weights-conductances dependency is proportional in a strictly monotonously manner (see Equation (4.3) and Equation (4.4)). The restriction to only positive weights yields an ANN classifier whose output function is intrinsically monotonic in relation to its inputs. This would limit the applicability to potential classification or regression problems.

In order to achieve negative weights, an inverter-based negative weights operation (inv) (Figure 4.4a) is proposed, which turns positive neuron input voltages into negative voltages and vice versa, similar to the operation  $x_i \times (-1)$  (or  $V_i \times (-1)$ ). The benefit compared to other existing techniques [62] is that the negative weights circuit is only used when necessary and then placed and inkjet-printed before crossbar resistors whose weights should be negated (Figure 4.6). Consequently, the resulting printed neuron design offers less area, significantly less power consumption and reduced transistor count. Moreover, less material has to be printed which also reduces printing costs and time.

The proposed circuit schematic of the negative weight operation is shown in Figure 4.4a. Due to the absence of p-type transistors in this technology, the pull-up network of the inverter consists of a resistor ( $R_5$ ) and an EGT in the pull-down network ( $T_1$ ). Moreover, two voltage dividers ( $R_1, R_2$  and  $R_3, R_4$ ) are inserted to improve the rail-to-rail behavior. The additional bias pin was inserted for fine-tuning and shifting of the zero crossing point but was however not required in the final experiment.

The resistor and transistor sizing was performed using SPICE simulations based on a prior developed printed process design kit (PPDK) [27]. The design parameter values are provided in Table 4.1. VSS was adjusted from initial  $-2.0V$  to  $-2.2V$  to shift the zero crossing point of the circuit transfer function towards  $0V$ . The output transfer function's range of values is approximately between  $-1V$  and  $1V$ , thus signal degradation across this circuit element is prevented as the full input voltage swing is also provided at the output.

The DC measurements of the negative weights circuit are shown in Figure 4.4c, where the input signal  $V_{in}$  is swept from  $-1V$  to  $1V$ . It can be seen that the output signal  $V_{out}$  follows the input signal  $V_{in}$  inversely proportional. As it is shown later, this behavior is adequate to approximate negative ANN weights which allow for successful classification of benchmark datasets. As can be obtained from Figure 4.4c, the output waveform of the fabricated hardware prototype deviates from the circuit simulation result. As the transfer function of the negative weights circuit is also not a perfect linear operation, and in order to propagate the imperfections

#### 4. Inkjet-Printed Neuromorphic Architectures

of the hardware prototype to the ANN learning method (see Section 4.1.5), a parameterized model of a true tanh was fit to the measured waveform, as depicted in Figure 4.4c, and used later during the simulation-based NN inference test (see Section 4.1.5). The used fitted model is:

$$\text{inv}(x) = -(a + b \cdot \tanh((x - c) \cdot d)), \quad (4.8)$$

with

$$a = 0.072, \quad b = 0.82, \quad c = 0.062, \quad d = 5.52$$

The choice of design parameters and the measurement results are depicted in Table 4.1. The reason why the power consumption of this circuit is so high is due to the first voltage divider  $R_1$  and  $R_2$ . As the resistances of both resistors are in the range of only hundreds of ohms and as secondly a constant voltage source is applied to them (VDD-VSS), a high current is induced which leads to high power dissipation. One way to achieve substantial power consumption reduction is by scaling all resistances  $R_1$ - $R_4$  up by a constant factor  $s \gg 1$ .

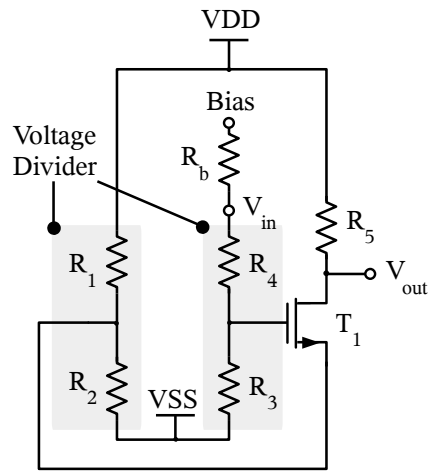
#### Tanh-like Activation Function

One problem with the neuron design and crossbar-based MAC operation proposed in Section 4.1.2, is that the absolute value of the weights is always smaller or equal to 1:  $w_i, w_b \in [0, 1]$ . This implies high signal losses at each ANN layer and consequently the output signals of the crossbar - the result of the MAC operation - becomes susceptible to signal noise. Second, further signal deterioration is induced by the pPLU, which reduces the positive input voltages to about 70% (e.g. 1V at the input of the pPLU will be converted to 700mV at the pPLU output).

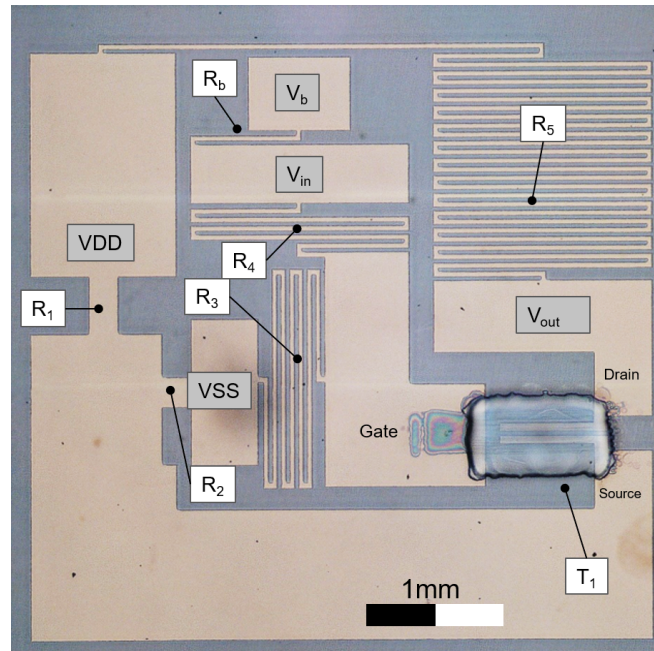
This voltage degradation can be compensated by a more suitable choice of an activation function, which despite its non-linear property also behaves as a voltage buffer element. Such an activation function can be realized by following an inverter-based approach, as depicted in Figure 4.5a. By cascading of two inverters, the circuit acts as a non-inverting buffer element, which restores voltage levels between the ANN layers. The behavior of this circuit resembles a hyperbolic tangent (tanh), thus it is abbreviated in the following as ptanh (printed tanh). The transfer function of the circuit is depicted in Figure 4.5c. In contrast to the pPLU, the resistors in the tanh-like activation function can be chosen independently to the resistors in the MAC circuit due to the high input impedance at the EGT gate terminal (EGT gate currents are typically in the order of nano Ampere).

For the ptanh hardware prototype, resistor sizing was achieved by design extraction based on SPICE simulations and the PPDK [27]. A microscopic photo is shown in Figure 4.5b. Two EGTs were printed, one for each inverter, and the conductive tracks were again obtained from laser-structured ITO-sputtered glass substrates (Section 2.4). The design parameters chosen for this component are illustrated in Table 4.1. The additional bias pin for zero crossing point tuning was not required in this experiment.

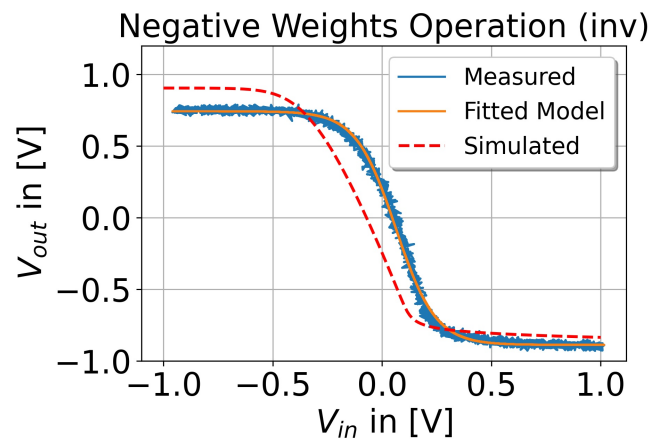
The input and output waveforms are shown in Figure 4.5c. As can be seen, the zero crossing point is close to 0V input voltage. Moreover, excellent rail-to-rail behavior is observed. The



(a) Schematic



(b) Microscopic Photo



(c) Measured Waveforms

**Figure 4.4.:** (a) shows the schematic of the proposed negative weight circuit. Two voltage dividers are deployed to shift the zero-crossing of the output voltage towards 0V. (b) depicts the microscopic photo of the hardware prototype. (c) contains the simulated and measured transfer function of the circuit as well as the fitted model.

#### 4. Inkjet-Printed Neuromorphic Architectures

output signal voltage levels are ideally pulled up/down to  $1V/-1V$ , thus this component can be preferably deployed as a voltage buffer for voltage signal replenishment at the output of each ANN layer. It was found experimentally that the sensing resolution of the output signals of this circuit is about  $100mV$ . Thus, in order to distinguish output signals and predict the classification outcome correctly, the difference between output signal voltage levels must be at least  $100mV$ . The ANN training routine was made aware of this constraint by introducing a safety margin during the ANN learning phase (see Section 4.1.5).

Similar for the negative weights circuit, there is a mismatch between the simulated behavior and the measured output waveform (Figure 4.5c). Thus, a model was fit to the measured waveforms which is then deployed in the ANN learning method (see Section 4.1.5). This model is based on a parameterized version of a true tanh:

$$ptanh(x) = a + b \cdot \tanh((x - c) \cdot d),$$

with

$$a = 0.046, b = 1.0, c = 0.054, d = 9.11$$

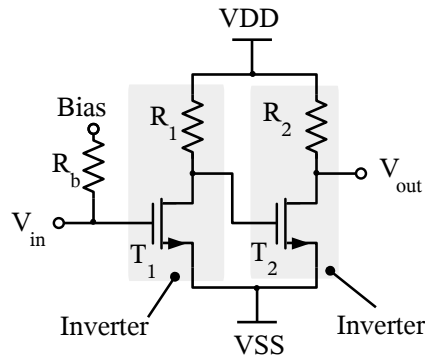
The fit is also indicated in Figure 4.5c. The measured delay, power consumption and area usage are presented in Table 4.1.

#### 4.1.4. Proposed ANN Hardware Architecture

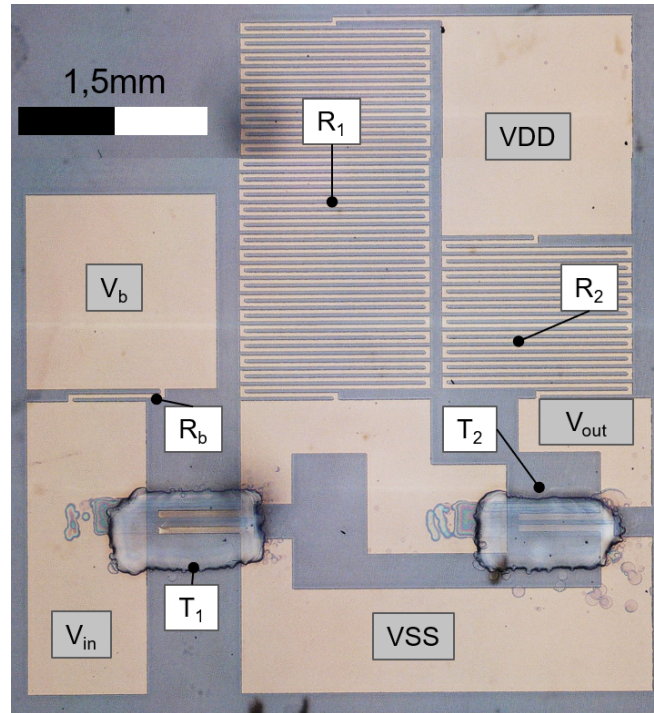
By interconnection of the three ANN building blocks - MAC, inv, ptanh - a printed functional neuron (pNeuron) can be constructed, with arbitrary number of neural inputs (i.e. synapses). It is remarkable that the pNeuron requires only two transistors for the activation function, and one transistor for each negative weight operation. This leads to very low hardware footprint and small delay and power consumption in comparison with a conventional digital implementation.

These benefits can also be derived from Table 4.3, where an analog 3-input neuron was compared simulation-based with a 4-bit (low-precision) and 8-bit (high-precision) digital implementation. The digital neuron deploys a fixed-point multiplication unit with a rectified linear unit (ReLU) as the activation function. In order to keep the number of transistors reasonable, the multiplications and additions of the digital 3-input neuron are performed sequentially. For the digital components, high-level synthesis tools were deployed to extract circuit characteristics [63]. Although the digital neuron processes the inputs sequentially, a large number of transistors are required due to the implementation of expensive ADCs and multiplier units. Thus, these digital neuron designs are infeasible to be fabricated by the inkjet-printing technology. When increasing the precision of the digital circuits from 4-bit to 8-bit, a trend of exponentially increasing transistor count, area, delay and power consumption can be observed. Moreover, as can be obtained from Table 4.3, the analog implementation requires much less area and is also superior in terms of delay and power consumption in comparison to the 4-bit and 8-bit digital neuron implementation. Obviously, this analysis encourages the utilization of analog designs in PE for NCS realization.

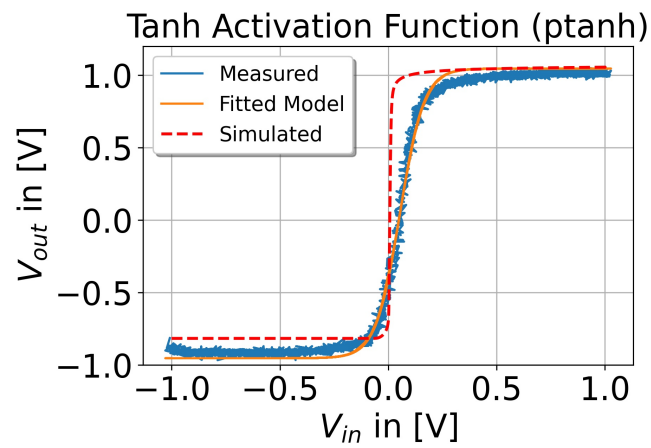
Based on the pNeuron design, larger NNs can be constructed by replicating the proposed neurons intra- and inter-layer-wise (Figure 4.6 top).



(a) Schematic



(b) Microscopic Photo



(c) Measured Waveforms

**Figure 4.5.:** (a) shows schematic of the inverter-based activation function for realizing the tanh function (b) depicts a microscopic photo of the fabricated hardware prototype of the ptanh (c) contains the simulated and measured waveforms as well as the fitted model.

**Table 4.3.:** Comparison between proposed ANN components in EGT-Technology (ADC: Analog-Digital-Converter, ReLU: Rectified Linear Unit) and conventional digital implementation (with 4-bit and 8-bit precision). The underlying neuron design for both the digital and analog implementation has 3 inputs. In the case of the digital implementation (4-bit and 8-bit), the computations (addition and multiplication) are performed sequentially.

Precision	Components	Delay	Area	Power	#Transistors
4-bit	ADC	13.8ms	25.4mm <sup>2</sup>	328μW	185
	Adder	13ms	7.9mm <sup>2</sup>	289μW	59
	Multiplier	13.6ms	15mm <sup>2</sup>	550μW	103
	ReLU	2.5ms	1.7mm <sup>2</sup>	80μW	10
	Neuron	69ms	48mm <sup>2</sup>	1.25mW	357
8-bit	ADC	154ms	957mm <sup>2</sup>	37.18mW	5938
	Adder	29ms	22mm <sup>2</sup>	793μW	144
	Multiplier	28ms	85mm <sup>2</sup>	3.1mW	583
	ReLU	2.55ms	3.7mm <sup>2</sup>	210μW	22
	Neuron	522ms	1068mm <sup>2</sup>	41.25mW	6602
<b>Analog</b>	<b>Neuron</b>	<b>27ms</b>	<b>0.49mm<sup>2</sup></b>	<b>859μW</b>	<b>4</b>

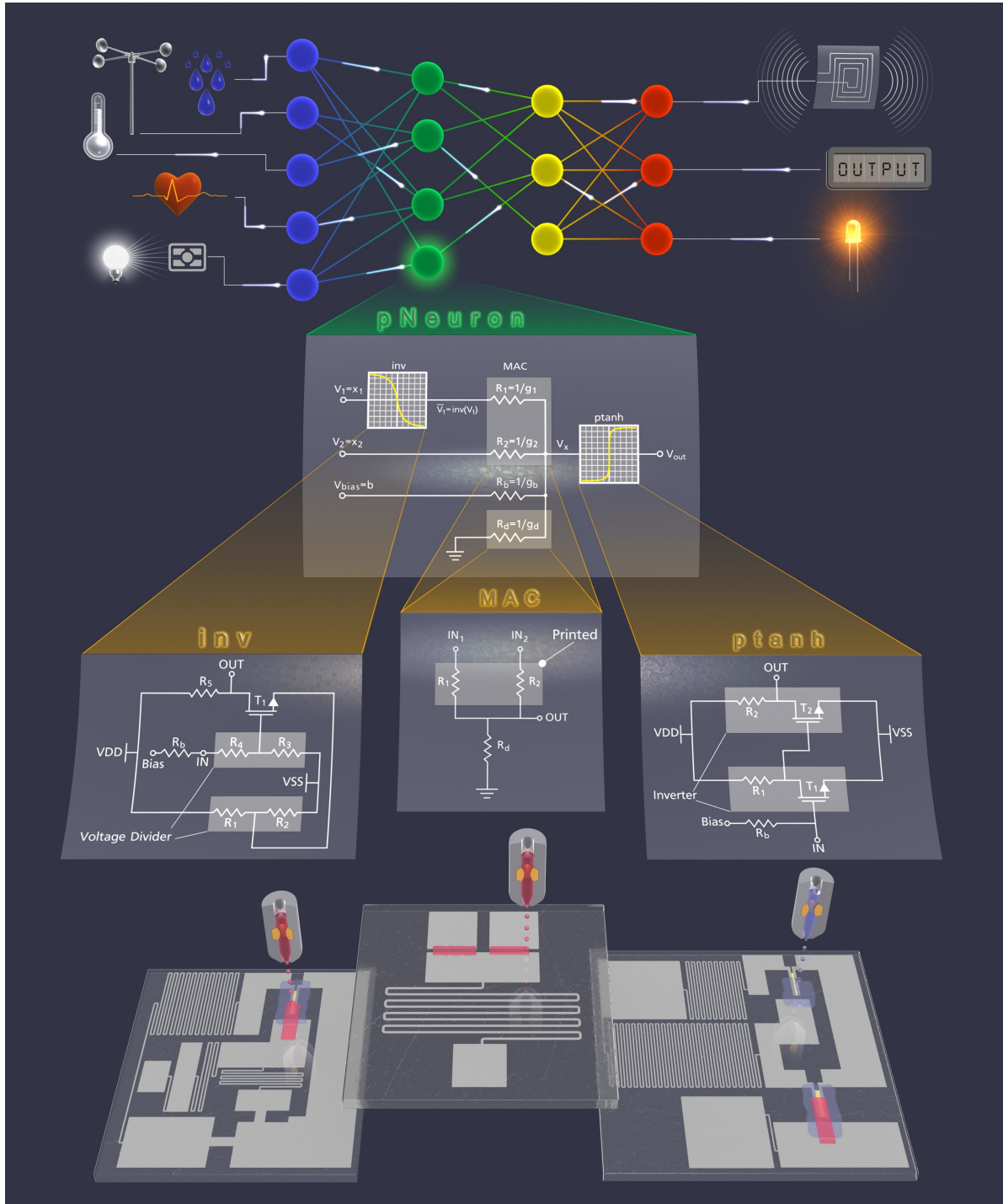
At the top of Figure 4.6, the high-level perspective of a printed NCS is illustrated. The printed NCS can be placed next to a sensor, from which it collects the input data. The sensory data is processed by the printed NCS using ANN inference. The ANN output signals can interface actuators for control commands, printed displays or wireless communication devices, to name a few examples.

The resulting ANN can be deep with many hidden layers due to the amplification of the ANN outputs at the non-linear activation function. Furthermore, the printed NCS can be tailored to a target application, by first choosing the ANN topology, number of nodes and layers, and second by programming the MAC operation by resistor printing to the crossbar interconnects after ANN training for point-of-use customization. As the negative weights circuit is only inserted at the required locations, this leads to a sparse ANN implementation with fewer components compared to existing designs [62].

#### 4.1.5. Training of Printed ANN

As ANN learning routines are tailored to the functional behavior of MAC operations and activation functions, dissimilarities caused by different technologies prevent the application of existing learning solutions [62] to converge. In addition, the customization feature of inkjet-printed NNs allow to explore new ANN topologies such as sparse ANN interconnections and low-device-count MAC operations. This yields in gains in ANN size and speed, as well as simplifications in ANN learning and inference. For this reason, existing hardware-agnostic





**Figure 4.6.:** *Top:* Printed neural network with sensor inputs and actuators, *middle:* circuit schematics of the inkjet-printed neuron (pNeuron) and the hardware prototypes of its three fundamental building blocks: negative weights circuit (inv), multiply accumulate operation (MAC) and non-linear tanh-like activation function (ptanh) *bottom:* physical layout of the hardware prototypes

#### 4. Inkjet-Printed Neuromorphic Architectures

training algorithms must be adapted to consider technology-related particularities.

For instance, the conductivity of an inkjet-printed resistor is bounded:  $g_i \in [g_{min}, g_{max}] \cup \{0\}$ . The lower and upper bounds  $g_{min}, g_{max}$  are due to the constant resistivity of the printed conductive ink as well as constraints on the lateral geometries as a result of limited area. The conductance of 0 is achieved by not printing a resistor at all, which leads to a conductance of a few pico Siemens due to the very low substrate currents. Another constraint is that negative weights are obtained by the inv-circuit, which is not a perfect linear function with negative slope. Finally, the voltage signals of the printed ANN output nodes must be measurable and distinguishable to obtain the correct classification outcome. These constraints are addressed during the ANN training routine as follows.

To simplify the training algorithm, the ANN is trained with respect to surrogate conductances  $s_i$ , which can be negative and positive real-valued numbers. The resulting ANN weights are obtained from the surrogate conductances by:

$$w_i = \frac{|s_i|}{\sum_j |s_j| + |s_b| + |s_d|}. \quad (4.9)$$

The output of the MAC function incorporates both the surrogate conductances and the inv-circuit to achieve the negative weights operation. The MAC output used during training which maps the surrogate conductances and input voltage signals to the crossbar output is:

$$\sum_i w_i (x_i \cdot \mathbb{1}_{\{s_i \geq 0\}} + \text{inv}(x_i) \cdot \mathbb{1}_{\{s_i < 0\}}), \quad (4.10)$$

where  $\text{inv}()$  is the fitted inv-circuit transfer function (Figure 4.4c and Equation (4.8)) and  $\mathbb{1}_{\{\cdot\}}$  is the indicator function which returns 1 if the argument is positive, and 0 otherwise.

For the ANN learning, back-propagation is used [59] and a noise margin for signal separation at the ANN output layer is guaranteed by a custom loss function with a penalty term. Here, a loss function is constructed inspired by the multi-class-hinge-loss [64]:

$$L(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}} l(\mathbf{x}, y, \boldsymbol{\theta}),$$

with

$$l(\mathbf{x}, y, \boldsymbol{\theta}) = (m + T - f_{\boldsymbol{\theta}}(\mathbf{x})_y)^+ + (m + \max_{j \neq y} f_{\boldsymbol{\theta}}(\mathbf{x})_j)^+,$$

where

$\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=0}^N$  is the training data, and  $\boldsymbol{\theta}$  denotes the weights and biases of the ANN.  $f_{\boldsymbol{\theta}}(\mathbf{x}_n)$  is the ANN output vector given an input sample  $\mathbf{x}_n$ .  $f_{\boldsymbol{\theta}}(\mathbf{x})_y$  is the voltage at the node corresponding to the true label, while  $f_{\boldsymbol{\theta}}(\mathbf{x})_j$  denotes all other output nodes voltages.  $T \in \mathbb{R}^+$  is a user-defined threshold value and  $m \in \mathbb{R}^+$  is the margin which separates the correct classification output from the other output nodes, and  $(\cdot)^+ = \max\{0, \cdot\}$ .

By using this loss function, a positive loss only occurs when  $f_{\boldsymbol{\theta}}(\mathbf{x})_y$  is not higher than  $m + T$  and any  $f_{\boldsymbol{\theta}}(\mathbf{x})_j$  is bigger than  $-m$ . Thus, this loss function rewards an output vector where the correct output label is  $2m$  separated from all other outputs and furthermore increased by  $T$  for better robustness and differentiable voltage signal.

The learning routine was initialized by surrogate conductances distributed around 0 to avoid vanishing gradients induced by the ptanh saturation region in the first cycles of the backpropagation. Moreover,  $s_b$  was used as a tuning parameter to shift the neuron activation towards 0V to further avoid drifting into saturating regions of the activation functions.

In order to guarantee that the trained and obtained ANN weights are in the feasible range, which is:  $[-g_{max}, -g_{min}] \cup \{0\} \cup [g_{min}, g_{max}]$ , the obtained surrogate conductances were first projected to  $[-g_{max}, g_{max}]$  using clipping during the weight update step of the back-propagation algorithm. Second, an additional penalty term is added to the loss function, which penalizes the regions  $[-g_{min}, 0]$  and  $[0, g_{min}]$ . After the final weights from the training routine are obtained, the surrogate conductances which are still in the range  $[-g_{min}, 0]$  and  $[0, g_{min}]$  are rounded to 0.

#### 4.1.6. Benchmark Results

For the validation of the printed ANN concept and adapted training algorithm, an evaluation was carried out on nine benchmark datasets obtained from the UCI ML repository [65]. The features were normalized to the range  $[0, 1]$  and a test/train split of 67%/33% was used. For the evaluation metric a measuring-aware accuracy was deployed, which is defined as:

$$\frac{1}{|D|} \sum_{(\mathbf{x}, y) \in D} \mathbb{1}_{\{i=y\}} \cdot \mathbb{1}_{\{f_{\theta}(\mathbf{x})_y > T\}} \cdot \mathbb{1}_{\{\forall j \neq i \mid f_{\theta}(\mathbf{x})_j < 0\}},$$

with  $i = \operatorname{argmax}_j f_{\theta}(\mathbf{x})_j$  and  $D$  denoting the evaluation data. Using this evaluation metric, not only the correct classification outcome is rewarded, but also if the correct output label is bigger than  $T$  and all other outputs are negative. This ensures that the outcome is measurable. For the evaluation and training,  $T$  was set to 100mV, while  $m$  was set to 50mV during training.

The networks were implemented using *pytorch* [66]. Random guess was used as the baseline, which always predicts the class which was the most frequent one during training. Also a hardware-agnostic ANN (reference NN) with same ANN topology as the printed ANN and true tanh instead of ptanh was evaluated.

The inference results are provided in Table 4.4. From the evaluation, two observations can be made. First, the accuracy of the trained printed ANN was always higher than the random guess. Second, the accuracy was for the majority of the datasets comparable to the reference NN, and for few datasets even superior. Finally, the fact can be derived that the printed ANN can successfully solve popular classification problems.

#### 4.1.7. Summary

Overcoming the von-Neumann-bottleneck by using synapses and neurons instead of storage elements and transistors, is one of the integral ideas behind neuromorphic or in-memory computing. This enables circuit designs, which have much lower area overhead compared to digital computing architectures. Especially designs for PE benefit from this to provide low-complex and easily manufacturable NCS for future application domains. For NCS, materials and devices, which embody ANN functions have to be developed together [67]. Thus, in this chapter a comprehensive concept for printed NCS was presented, including the technology,

	Dataset	Architecture neurons/layer	pNN		reference NN		Random Guess
			Train	Test	Train	Test	
	Acute Inflammations	6-4-3-2	1	1	1	1	0.475
	Balance Scale	4-4-3-3	0.9522	0.9179	0.8493	0.8454	0.4396
	Breast Cancer Wisconsin	9-4-3-2	0.9786	0.9481	0.9722	0.9784	0.6667
	Energy efficiency (y1)	8-4-3-3	0.8424	0.8268	0.821	0.7953	0.4331
	Energy efficiency (y2)	8-4-3-3	0.8541	0.8622	0.8074	0.7559	0.4646
	Iris	4-4-3-3	0.97	0.94	0.96	0.92	0.28
	Mammographic Mass	5-4-3-2	0.7621	0.761	0.7278	0.7327	0.5503
	Seeds	7-4-3-3	0.9143	0.9143	0.9071	0.8571	0.2714
	Tic-Tac-Toe Endgame	9-4-3-2	0.9875	0.9716	0.9875	0.9748	0.6404
	Vertebral Column (2 classes)	6-4-3-2	0.7923	0.767	0.7826	0.8155	0.6893
	Vertebral Column (3 classes)	6-4-3-3	0.7005	0.7379	0.7391	0.7476	0.5146

**Table 4.4.:** *The table contains the evaluation of the benchmark classification task by deployment of the proposed printed ANN architecture (pNN), the hardware-agnostic ANN (reference NN) and a random guess method, which predicts the classification outcome based on the most frequent class occurred during the training cycle.*

circuit design and learning rule to realize NCS fully in PE and enabling applications such as near-sensor processing or new architectures for IoT sensors, soft robotics and other smart products for the low-cost consumer markets.

All the designs of the three building blocks presented in this section, have the capability to process analog data, encoded by voltage levels. These physical quantities with their continuous value representation are a substitution of digital multi-bit logic, which would lead to infeasible design overheads in PE. Moreover, the presented building blocks have very low device count, not more than two transistors per device, which is even exceptionally in the analog electronics domain. The proposed circuit designs for this technology provide all functional operations to implement non-spiking artificial neural networks (ANNs). Several constraints from the technology and printed neuromorphic architecture are considered, such as limited set of digital and analog components in PE, and signal degradation across ANN layers. Moreover, the proposed neuron can be one-time programmed similar to the proposed LUT (see Section 3.2).

A significant advantage of the proposed approach is that arbitrary large printed (deep) neural networks can be built due to the signal restoration property of the printed neuron concept. The ANN is also scalable with respect to the ANN layer size due to the crossbar-based MAC operation. Besides the proposed ANN circuit implementations, an adaptive learning scheme was provided, which takes the technology-related particularities into account, such as the negative weights operation, tanh-like activation function and restricted weights representation. Finally, the proposed ANN architecture is validated simulation-based on benchmark datasets of popular classification problems.

The area requirement of the ANN in this section was about  $400\text{mm}^2$  with a delay of 30ms during ANN inference. However, extreme improvements in area usage, delay and power consumption are expected in future designs, which are beyond the capabilities of the presented hardware prototypes in this work. This work is considered as an important step towards fully inkjet-printed NCS for sensor processing applications.

## 4.2. Inkjet-Printed Stochastic Computing Neural Networks

Another low-area technique for reducing complexity of circuit designs is stochastic computing (SC). As the signals in SC are encoded by streams of random bits, complex operations such as addition and multiplication can be performed bit-wise compared to multi-bit representations used in digital computing. This reduces hardware footprint and wiring costs of SC components. For instance, a multiplier in SC can be realized by a single XNOR gate with eight transistors, compared to hundreds of transistors for a low-precision multiplier using digital computing.

However, SC has not experienced wide applicability in silicon-based electronics as performance and throughput are reduced due to bit-wise sequential processing. Moreover, area requirements are not a concern in silicon due to the small feature sizes which are in the nanometer range. On the other side, high-performance is not a primary requirement of PE applications. For instance, sensor readouts occur only periodically every hundreds of milliseconds [68, 69], seconds [70, 71] or even minutes [72] for many applications. In addition, as feature sizes in PE are in the micrometer range, area is a major concern.

SC was previously utilized for real-time image processing [73], digital filters [74] or implementation of polynomial functions [75]. Also in the context of ANNs, SC-based designs were developed [76, 77], which incorporate all functionalities for neuron implementation, such as MAC operations, activation functions and stochastic number generators (SNG), the latter required for conversion of real-valued signals into the SC domain.

However, as research on SC-based implementations was mainly based on silicon technology, no investigations were made in the PE domain. As it is shown in this section, while adders and multipliers can be implemented by a few transistors, other components required for SC-based neural network (SC-NN) hardware, such as activation functions and stochastic number generators require too many logic gates and power consumption, and are impractical for PE applications. To this end, implementations of analog printed activation functions and SNGs are presented, which require only a small amount of area and power compared to conventional digital realizations. Another benefit of the presented analog circuit is the capability to directly interface analog sensors by converting analog input signals directly to the SC-domain, without resorting to expensive digital ADCs.

The proposed mixed-signal SC-based neural network is the first study of stochastic computing for PE. In this section, an evaluation of conventional and proposed stochastic computing neural networks for PE is provided on the design and simulation level. Finally, the proposed SC-NN architecture is validated on popular benchmark dataset.

### 4.2.1. Stochastic Computing

The basic idea of stochastic computing is to encode continuous values as a sequence of random bits [78]. For instance, the real value of a stochastic number encoded in a bit stream of length  $L$  and bit values  $S_i$ , can be obtained by

$$Y = \frac{1}{L} \sum_{i=1}^L S_i. \quad (4.11)$$

#### 4. Inkjet-Printed Neuromorphic Architectures

As the bits  $S_i$  are random variables,  $Y$  is also stochastic. The precision of a stochastic number can be improved by increasing  $L$ , and the output  $Y$  approaches asymptotically the expected value  $\mathbb{E}[Y]$ . The number of real values, which can be represented by  $Y$  is  $L$ . E.g. the lowest number is 0 (0 '1's in the bit stream), and the highest is 1 (all bits are '1's). A number in between might be 5/16 (5 '1's with a bit stream length of  $L = 16$ ). From Equation (4.11) it is obvious, that  $Y$  lies in the interval  $[0, 1]$ . However, a stochastic number can also be interpreted as a bi-polar number [79] by using the following transformation:

$$\tilde{Y} = 2 * Y - 1, \quad (4.12)$$

which maps the values range to  $[-1, 1]$ .

The 2-input addition operations in SC can be realized by a single 2-input multiplexer [78]. The classical multiplexer Boolean logic function can be expressed as

$$Y = (A \wedge S) \vee (B \wedge \bar{S}), \quad (4.13)$$

where  $A$  and  $B$  are the multiplexer inputs and  $S$  is the multiplexer select signal.

The expected value of  $Y$ , when the input signals and multiplexer select signals are uncorrelated, can be computed by:

$$\mathbb{E}[Y] = \mathbb{E}[A] \cdot \mathbb{E}[S] + \mathbb{E}[B] \cdot \mathbb{E}[1 - S] \quad (4.14)$$

$$= \mathbb{E}[A] \cdot \frac{1}{2} + \mathbb{E}[B] \cdot \frac{1}{2} = \frac{1}{2}(\mathbb{E}[A] + \mathbb{E}[B]), \quad (4.15)$$

where the expected value  $\mathbb{E}[S]$  was set to  $\frac{1}{2}$  (i.e. a stochastic bitstream is used for the multiplexer select signal which has in average 50% '1's and 50% '0's). This formula holds also for bi-polar encoded stochastic numbers [77]:

$$\mathbb{E}[\tilde{Y}] = 2 \cdot \mathbb{E}[Y] - 1 = 2 \cdot \left[ \frac{1}{2}(\mathbb{E}[A] + \mathbb{E}[B]) \right] - 1 \quad (4.16)$$

$$= (\mathbb{E}[A] + \mathbb{E}[B]) - 1 \quad (4.17)$$

$$= \frac{1}{2}(2 \cdot \mathbb{E}[A] + 2 \cdot \mathbb{E}[B] - 2) \quad (4.18)$$

$$= \frac{1}{2}(2 \cdot \mathbb{E}[A] - 1 + 2 \cdot \mathbb{E}[B] - 1) \quad (4.19)$$

$$= \frac{1}{2}((2 \cdot \mathbb{E}[A] - 1) + (2 \cdot \mathbb{E}[B] - 1)) \quad (4.20)$$

$$= \frac{1}{2}(\mathbb{E}[\tilde{A}] + \mathbb{E}[\tilde{B}]). \quad (4.21)$$

The multiplication operation for bi-polar stochastic numbers can be realized by an XNOR gate [76]. Similar to the add operation, the Boolean function for the XNOR formula can be expressed logically:

$$Y = (A \wedge B) \vee (\bar{A} \wedge \bar{B}). \quad (4.22)$$

By applying the expected function to this logic expression, the following formula is obtained:

$$\mathbb{E}[Y] = \mathbb{E}[A] \cdot \mathbb{E}[B] + \mathbb{E}[1 - A] \cdot \mathbb{E}[1 - B] \quad (4.23)$$

$$= \mathbb{E}[A] \cdot \mathbb{E}[B] + (1 - \mathbb{E}[A]) \cdot (1 - \mathbb{E}[B]) \quad (4.24)$$

$$= 2 \cdot \mathbb{E}[A] \cdot \mathbb{E}[B] + 1 - \mathbb{E}[A] - \mathbb{E}[B] \quad (4.25)$$

$$= (2 \cdot \mathbb{E}[A] \cdot \mathbb{E}[B] - \mathbb{E}[A]) - \mathbb{E}[B] + 1 \quad (4.26)$$

$$= 2 \cdot \mathbb{E}[A] \cdot (\mathbb{E}[B] - \frac{1}{2}) - \mathbb{E}[B] + 1 \quad (4.27)$$

$$= 2 \cdot \mathbb{E}[A] \cdot (\mathbb{E}[B] - \frac{1}{2}) - (\mathbb{E}[B] - \frac{1}{2}) + \frac{1}{2} \quad (4.28)$$

$$= (2 \cdot \mathbb{E}[A] - 1) \cdot (\mathbb{E}[B] - \frac{1}{2}) + \frac{1}{2} \quad (4.29)$$

$$\Leftrightarrow 2 \cdot \mathbb{E}[Y] - 1 = (2 \cdot \mathbb{E}[A] - 1) \cdot (2 \cdot \mathbb{E}[B] - 1) \quad (4.30)$$

$$\mathbb{E}[\tilde{Y}] = \mathbb{E}[\tilde{A}] \cdot \mathbb{E}[\tilde{B}]. \quad (4.31)$$

#### 4.2.2. Related Work on SC-based NNs

An SC-NN is based on the same operations used for conventional ANN architectures: MAC operations and activation functions. Also ANN training and inference techniques are similar. However, there are a few key differences. First, the 2-input adder operation is always scaled by a factor of  $\frac{1}{2^{\lceil \log_2 N \rceil}}$ , where  $N$  is the number of adder inputs (e.g.  $\frac{1}{2}$  for 2 inputs, and  $\frac{1}{4}$  for four inputs etc.). Second, the latency of SC-NNs depends on the length of the stochastic bit-stream used to represent the stochastic encoded numbers in the SC-NN. For instance, when stochastic numbers are encoded with 32 bits, they are processed  $4\times$  faster than using 512 bits (however the latter enables higher ANN inference accuracy). Moreover, with an SC-NN, an early classification result can be obtained by evaluation of only the first arriving bits of the stochastic bit-stream at the ANN output nodes. As a result, a rough estimate of the classification outcome is obtained, whose precision is increased by evaluation of the successive bits, also termed as progressive precision.

Another difference is the limited range of values a stochastic number can represent, which is  $[0, 1]$  for unipolar encoding and  $[-1, 1]$  for bi-polar encoding (as discussed in Section 4.2.1). As a result, the value ranges of all signals such as inputs, weights and outputs are bounded to these intervals. As at each neuron, multiplications are applied and the inputs and weights are not larger than 1, the signal is degraded at each neuron and ANN layer. Computational range expansion is deployed in this context to allow stochastic numbers to have absolute values larger than 1 by using an integer form of stochastic computing [76] or by deployment of the ratio of stochastic numbers [80], however at the expense of additional hardware overhead.

Besides the MAC operation, the implementation of activation function in SC is non-trivial. Research efforts were made to implement silicon-based activation functions such as stochastic hyperbolic tangents using finite state machines (FSM) [77]. Also rectified linear units (ReLUs)

were implemented by accumulative parallel counter (APC) based FSMs [78]. Another approach deployed Taylor-series expansion to the ideal non-linear activation function and approximated the behavior by SC-based polynomial arithmetic circuits [78]. Although these techniques can be easily applied to conventional hardware, they can only partially be mapped to PE-based circuits due to the high area overheads.

### 4.2.3. Proposed SC Designs for PE

#### Motivation: Limitations of Printed Digital ANN and SC-NNs

In Figure 4.7 an artificial neuron is depicted implemented by digital computing as well as by SC. The SC components for the SNGs realized by True Random Number Generators (TRNGs), are provided as well as an SC-based activation function (AF) implemented by an FSM. In general, the SNGs are deployed for conversion of binary numbers into stochastic bit streams to realize ANN input features, weights or select signals to drive the multiplexers for the add operation.

In general, the MAC operation for ANN computations requires at least one multi-bit adder and a fixed-point multiplier. As can be derived from Figure 4.7, a conventional adder is built from XNOR gates and Full-Adder blocks, the latter requires many transistors. In addition, a digital multiplier contains several adders and additional combinational logic, which leads to even higher transistor count compared to the digital adder. As a result, a digital MAC operation implemented in PE is infeasible due to the complex designs. As an example, an 8-bit and 4-input MAC operation requires 1310 transistors (Table 4.5). Thus, a digital 8-bit neuron with 3 inputs in EGT technology has a delay of 243ms with an area usage of 3174mm<sup>2</sup>, and a power consumption of 123mW (see Table 4.6 - obtained based on synthesis results using EGT standard cell library [63]).

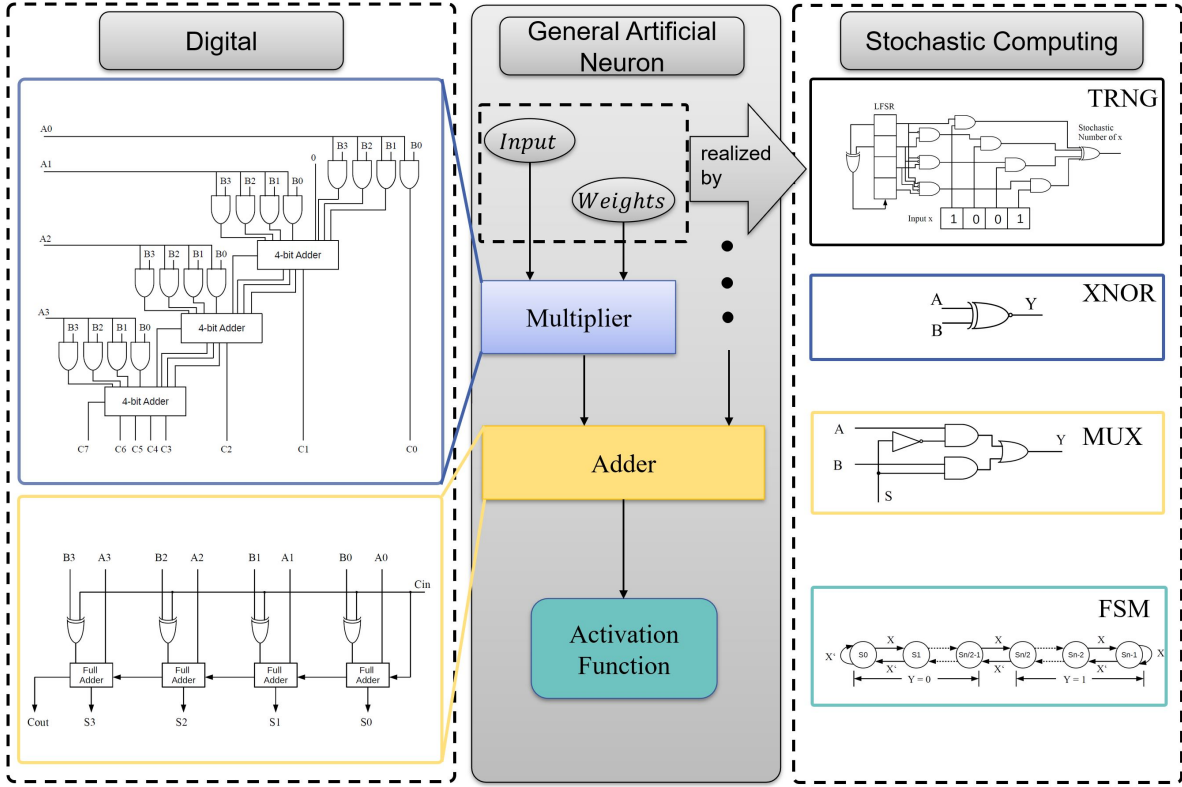
Obviously, a printed neuron based on a fully digital design cannot be implemented in PE or powered by an energy harvester system or lightweight battery. On the other hand, a MAC operation implemented in the SC domain, requires only 23 transistors (57× fewer) as the multiplication operation is replaced by a single XNOR gate, and the addition operation by a 2-input multiplexer (MUX) (see Figure 4.7). The resulting SC-based MAC operation has substantial lower area and power consumption making it feasible to be fabricated in PE.

However, although SC-based multipliers and adders have low transistor count, other circuitry deployed for ANN computations such as activation functions require still complex designs. The SC-based activation functions are usually implemented by FSMs (see Figure 4.7), which contain flip flops and additional combinational circuitry, leading to hundreds of transistors. Also SNGs have high device count, commonly realized by digital TRNGs. As SNGs are deployed in large numbers, for each ANN input, weights and multiplexer select signals, they dominate the total chip area (see Table 4.5) and diminish the area and power consumption gains obtained by the low-complexity SC-based adder and multiplier operations. To still reach feasible and printable SC designs, novel and inexpensive SNGs and activation function designs have to be explored.

#### Analog Components for printed SC-NN

Similar to the analog NCS presented in Section 4.1, circuit designs from the analog domain are explored for the implementation of the SC-based SNGs and activation functions, in order

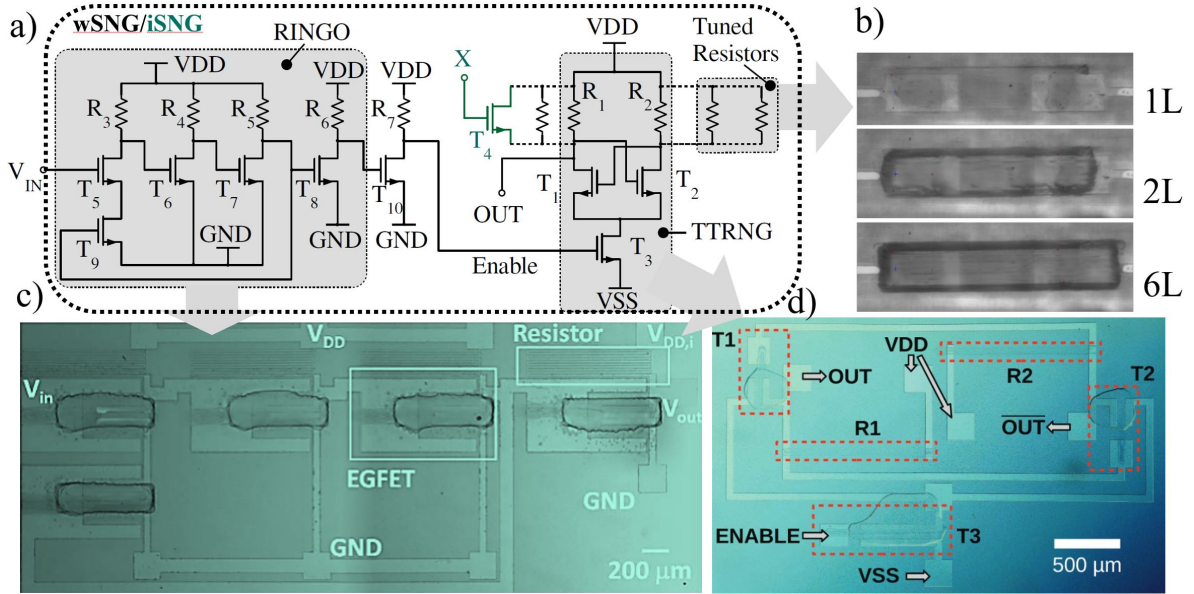




**Figure 4.7.:** Implementation of artificial neuron components using digital computing vs stochastic computing

to reduce the transistor count. The circuit design of the proposed analog stochastic number generator, which implements the ANN weights (wSNG) is illustrated in Figure 4.8. The functionality of the wSNG can be described as follows: first an oscillating signal is generated by a printed ring oscillator (RINGO) [81]. Subsequently, the oscillating signal is applied to the gate of a transistor ( $T_3$ ) which enables a tuned true random number generator (TTRNG). The TTRNG consists of a bi-stable back-to-back inverter ( $T_1, T_2, R_1, R_2$ ) which outputs either logic '1' or '0', depending on the ratio of the pull-up resistors  $R_1$  and  $R_2$ . In phase with the oscillating enable signal, a stochastic bit-stream is generated. It is important to note, that the TTRNG is tuned in a post-fabrication step by printing additional layers to the pull-up resistors [82]. This is possible, as each additional layer can be considered as a parallel connected resistor and thus the total resistance is decreased [82]. As a result, the probability of producing '0's and '1's can be controlled for the generation of different stochastic numbers required for the ANN weights. While post-fabrication tuning can be achieved in PE, it is not achievable in subtractive processes such as silicon-based technologies.

The wSNG can be extended to enable also the generation of input features for the SC-NN. By insertion of an additional transistor ( $T_4$ ) to the pull-up network (see Figure 4.8), analog ANN input voltages ( $X$ ) can be converted into stochastic numbers. This circuit is denoted in the following as iSNG. While the iSNG is input-controlled, the wSNG is one-time programmable. The benefit of using wSNGs is that no explicit weight storage is required [77], as the weights are encoded by printed resistors. This reduces the hardware footprint for SC-NNs even further. In summary, using iSNGs and wSNGs, all SC-based ANN signals can be generated, required for the full implementation of an SC-based ANN. The functionality of the SNG was validated using simulations as depicted in Figure 4.9.

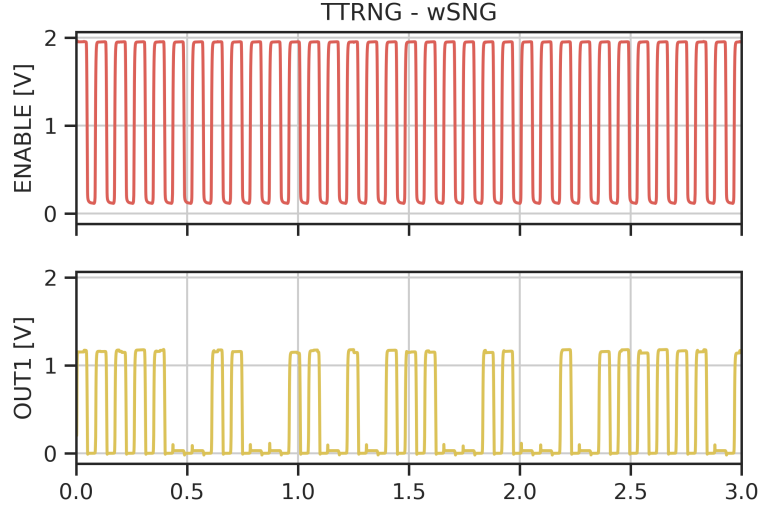


**Figure 4.8.:** Schematic and microscopic photos of proposed stochastic number generator: a) Schematic of the proposed stochastic number generator consisting of a ring oscillator and a tuned true random number generator (TTRNG) for wSNG, and additional transistor  $T_4$  (green) for iSNG. b) microscopic photo of resistor tuning in a post-fabrication step, by printing first 1 layer (1L), 2 layers (2L) and 6 layers (6L) of conductive materials c) microscopic photo of printed ring oscillator [81] d) microscopic photo of printed TRNG [82]

For the SC-based activation function, an analog design was derived, which is based on a capacitor for analog integration of voltage pulses, which are related to the incoming stochastic bit stream. Such a capacitor can be built by using the electrolyte-semiconductor interface, which are based on the same functional inks used for EGT fabrication. To obtain different capacitances, the area of this interface has to be adjusted, similar to conventional capacitors.

The circuit schematic of the proposed analog activation function is illustrated in Figure 4.10. The input stochastic bit stream is applied to the input stage at the input port (IN). Depending on the enable signal (EN), the activation function operates in two states. In the first operating state, the enable signal is logic '1' and the input voltage pulses charge (input pulse is logic '1') or discharge (input pulse is logic '0') the capacitor  $C_1$  via transistor  $T_2$ , or  $T_3$  respectively.

In the second operating state, when the enable signal is logic '0', the capacitor is disconnected from the charging stage and its voltage level is kept constant, proportionally to the ratio of '1's and '0's in the processed stochastic bitstream. Depending on the voltage at  $C_1$ , transistor  $T_6$  is either switched on or off. If  $T_6$  is switched on, the current voltage pulses from the incoming bitstream propagate through the output stage to the port OUT. If  $T_6$  is switched off, the output (OUT) is pulled down to ground (logic '0'). The functionality of this circuit is similar to a rectified linear unit (ReLU), where negative inputs lead to a constant output of 0V, and which behaves as a linear function for positive input values. With respect to the SC-based activation function, negative inputs (less than 50% '1's in the bitstream) are mapped to -1 (in bi-polar encoding '0's in a bitstream represent -1), while positive inputs (more than 50% '1's in the bitstream) propagate to the output, thus a linear function with slope '1' is provided at the activation function output. Due to this, the proposed analog SC-based activation function is denoted as a bi-polar rectified linear unit (bi-polar ReLU). In order to reset the analog



**Figure 4.9.:** Simulation of the printed analog wSNG. The following signals are depicted: *ENABLE*: the true random number generator (TRNG) enable signal. *OUT*: the random output bit stream with a length of 35 bits. In this example, as 23 bits are logic '1', the stochastic number presents 23/35.

integrator to its initial state to process the next incoming bitstream, a discharging transistor  $T_8$  was inserted which pulls the capacitor voltage to ground (0V) when switched on. The functionality of the bi-polar ReLU was validated using simulations as depicted in Figure 4.11.

### Overall Architecture

The SC-based neuron design is depicted in Figure 4.12. At the neuron input, features and weights are converted into stochastic numbers by the iSNGs and wSNGs, respectively. Multiplication between input features and weights are performed by the XNOR gate. The multiplication results are summed up by multiplexers. Both operations implement the MAC operation. As discussed in Section 4.2.1, the addition result is scaled down by a factor of 1/2 (or less dependent on the number of adder inputs). Finally, the bi-polar ReLU is applied to the adder results and the output is passed to the neuron input in the next layer or represents already the classification outcome when being part of the output layer.

### Training of proposed SC-NN

Similar to the printed analog ANN presented in Section 4.1, several technology-dependent constraints have to be taken into account during ANN training of the SC-NN.

First, the bi-polar encoded input features and weights are bounded to the interval (see Section 4.2.1):

$$x_i, w_i \in [-1, 1]. \quad (4.32)$$

Second, the addition operation scales the adder result by a factor of  $\frac{1}{2}$  (see Section 4.2.1). If more than two quantities are summed up, the addition operation becomes:

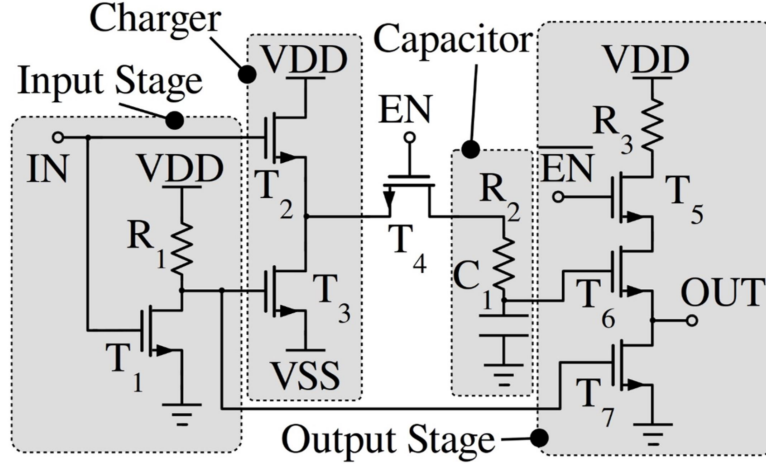


Figure 4.10.: Schematic of the analog activation function which resembles a bi-polar ReLU.

$$y = \frac{1}{2^{\lceil \log_2 N \rceil}} \sum_{i=1}^N x_i, \quad (4.33)$$

where  $x_i$  are the adder inputs,  $N$  is the number of inputs and  $y$  is the adder output.

These constraints are incorporated in the training procedure. The classical MAC operation is weighted by the scaling factor and the bi-polar ReLU is used instead of the conventional ReLU activation function in the training algorithm. Moreover, in each training cycle, the weights are mapped to the feasible range ( $w_i \in [-1, 1]$ ) using clipping. Thus, the weight update for step  $(t + 1)$  can be denoted as:

$$\vec{w}^{(t+1)} = \text{clip} \left( \vec{w}^{(t)} - \alpha^{(t)} \nabla \vec{w}^{(t)} \right), \quad (4.34)$$

where  $\alpha^{(t)}$  is the ANN learning rate,  $\vec{w}$  is the weights vector and

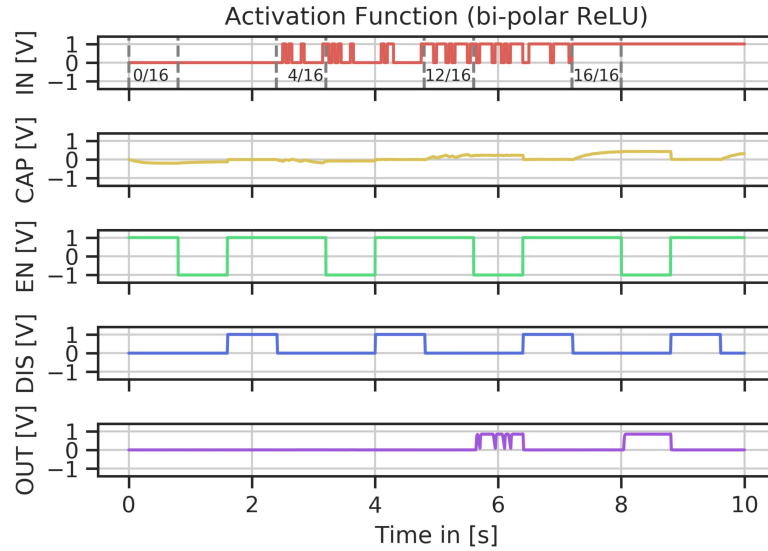
$$\text{clip}(z) = \begin{cases} -1 & z < -1 \\ z & z \in [-1, 1] \\ 1 & z > 1 \end{cases} \quad (4.35)$$

defines the clipping function.

### Discussion of Variations in Proposed Analog Designs

It is important to mention, that compared to digital designs, analog implementations are usually more susceptible to noise and variation. However, it was experimentally validated in previous work that the TRNGs of the printed SNG can be compensated and unbiased with respect to process variations in a post-fabrication step [82].

For the bi-polar activation function, simulation-based sensitivity analysis was performed, where process variations were injected into the circuit parameters to assess the impact on circuit functionality. In this experiment, all parameters such as transistor thresholds and resistor resistances were varied by 20% of their nominal values. Furthermore, the worst case



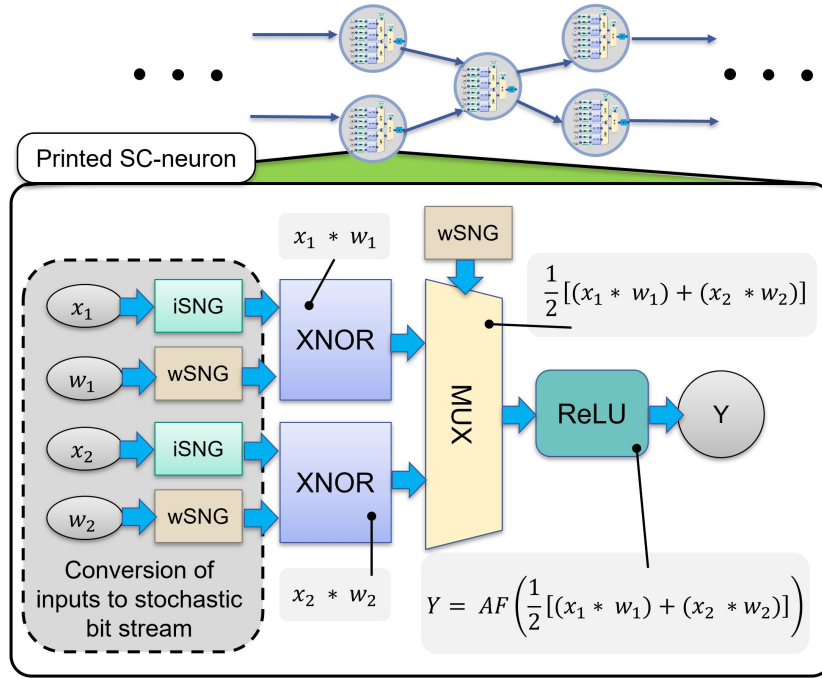
**Figure 4.11.:** *Simulation of the printed analog bi-polar ReLU. The following signals are depicted: IN: input bit stream, CAP: capacitor voltage, EN: enable signal, DIS: discharging signal, OUT: output bit stream. As can be obtained, only when the input bit-stream has more than 8 '1's out of 16 ( $> 8/16$ ), the input signal can propagate through the output.*

was considered, where the input bitstream contains 50% '1's and 50% '0's, which represents a bi-polar encoded stochastic number of 0, which is also the chosen threshold value of the activation function (point of inflection of the bi-polar ReLU). As the threshold of the activation function is determined by the capacitor voltage, the impact of component variations on the capacitor voltage also impacts the circuit functionality. Thus, the capacitor voltage was chosen as a quantitative measure for evaluation of the process variation impact. The sensitivity analysis experiment indicated that only four circuit components have very high correlation with the activation function functionality:  $C_1$ ,  $R_2$ ,  $T_2$ , and  $T_3$ . From Monte Carlo simulations carried out on these circuit parameters, it was however concluded that the capacitor voltage was only biased in average by 1mV, which corresponds to 2% of the maximum capacitor voltage during variation-free operation. Consequently, the bi-polar ReLU operated close to the variation-free case.

#### 4.2.4. Simulation Results

Compared to a digital SC implementation, the proposed analog designs improved area usage and power consumption substantially. As can be obtained from Table 4.5, the analog SNG requires only 2.5% of the area and 0.8% of power consumption compared to the digital 8-bit SNG. Similarly, the analog activation function, has only 17%/7% area usage/power consumption compared to the digital SC-based 8-bit components, however, at the expense of increased circuit delay.

The comparison is also performed on a higher-level, where the proposed mixed-signal approach is compared against a 4-bit and 8-bit digital implementation as well as a fully digital SC-based implementation. Evaluations are carried out for a single 3-input neuron (Table 4.6) and a full ANN (Figure 4.13) with 9 inputs, 3 hidden nodes, 2 output nodes (9-3-2).



**Figure 4.12.:** Printed stochastic computing neuron for SC-NNs with annotated mathematical behavior. The inputs  $x_i$  and weights  $w_i$  are converted into stochastic numbers using SNGs. Each input/weight pair is multiplied with the XNORs and finally added by the multiplexer, whose select signal is driven by a stochastic number as well. The result of the addition is then passed to the bi-polar ReLU activation function (AF). The printed SC-based neuron can be interconnected with other neurons to form large-scale neural networks with arbitrary topology.

From Table 4.6 it can be concluded, that the proposed mixed-signal neuron contains only 0.6% of transistors compared to an 8-bit digital SC-NN. As a result, the area requirement and power consumption are 0.6% and 0.6% respectively. As a reference, a conventional digital 8-bit neuron requires  $138\times$  more transistors to implement the same neuron. The low-precision 4-bit digital ANN contains  $6.9\times$  more EGTs than the proposed design.

A similar trend can be observed for the full ANN implementation (Figure 4.13). As can be derived from Figure 4.13, the improvements in transistor count for the proposed mixed-signal SC-NN are at the expense of higher inference time compared to the digital SC-NN. As the length of the stochastic bitstream impacts the inference delay, the performance of the SC-NN and digital ANN cannot be compared directly. As choosing long bitstreams improves the accuracy, it increases the delay on the other hand, and vice versa. In the following, the bit stream length was fixed to 1024 for the discussion on the inference results.

Table 4.7 shows the inference accuracy results for different design points. Thirteen benchmark datasets were chosen from the UCI ML Repository [65] for evaluation. The evaluated design points include a hardware-agnostic ANN (unconstrained weights, true ReLU), a PE specific ANN (weights bounded, stochastic adder (Equation (4.16)), bi-polar ReLU), deterministic 4-bit and 8-bit ANNs (fully digital), and the stochastic computing ANNs (random bit stream length of 1024) - both 4-bit and 8-bit digital implementations and mixed-signal SC (proposed) implementation.

**Table 4.5.:** Comparison between 4- and 8-bit digital components, 4- and 8-bit digital SC components and analog SC implementations. MAC operation has 4 inputs.

	Components	Delay	Area	Power	#EGTs
Digital 4-bit	ADDER	13ms	7.9mm <sup>2</sup>	289μW	59
	MULT	13.6ms	15mm <sup>2</sup>	550μW	103
	MAC	26.6ms	37.9mm <sup>2</sup>	1389μW	265
	AF	2.5ms	1.7mm <sup>2</sup>	80μW	10
	ADC	13.8ms	25.4mm <sup>2</sup>	328μW	185
Digital 8-bit	ADDER	29ms	22mm <sup>2</sup>	793μW	144
	MULT	28ms	85mm <sup>2</sup>	3100μW	583
	MAC	57ms	192mm <sup>2</sup>	6993μW	1310
	AF	2.55ms	3.7mm <sup>2</sup>	120μW	22
	ADC	154ms	957mm <sup>2</sup>	37 200μW	5938
Digital SC 4-/8-bit	ADDER	2.4ms	0.97mm <sup>2</sup>	33μW	7
	MULT	3.9ms	1.4mm <sup>2</sup>	51μW	8
	MAC	6.3ms	3.77mm <sup>2</sup>	135μW	23
	4-bit SNG	15ms	34mm <sup>2</sup>	5990μW	228
	8-bit SNG	23ms	71.7mm <sup>2</sup>	11 030μW	436
	4-bit AF	9.9ms	16.7mm <sup>2</sup>	1920μW	115
	8-bit AF	5.3ms	15.15mm <sup>2</sup>	1670μW	103
<b>Analog</b>	<b>SNG</b>	<b>50ms</b>	<b>1.76mm<sup>2</sup></b>	<b>94.53μW</b>	<b>11</b>
	<b>AF</b>	<b>50ms</b>	<b>2.62mm<sup>2</sup></b>	<b>116.7μW</b>	<b>8</b>

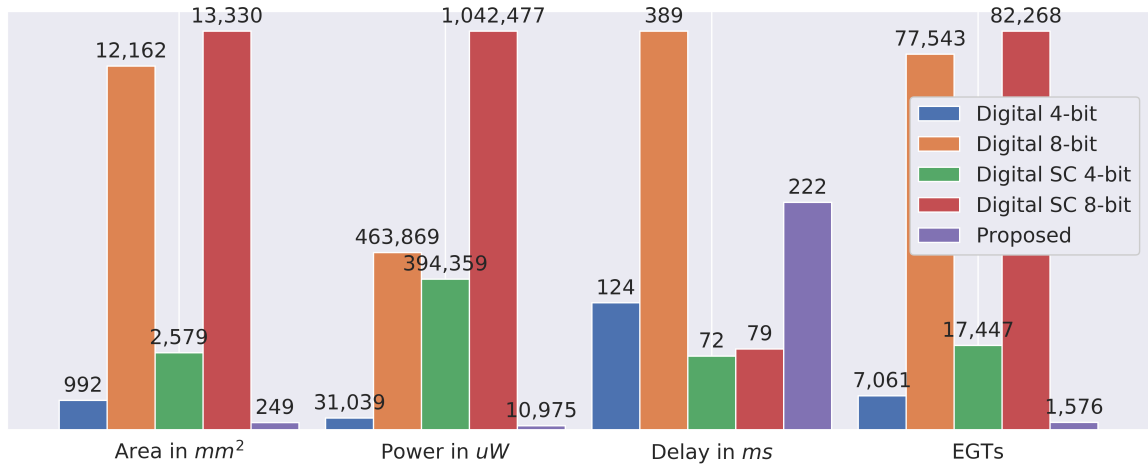
**Table 4.6.:** Comparison between digital 3-input artificial neuron for 4- and 8-bit conventional ANN, digital SC-based neuron for 4- and 8-bit digital SC-NN, and mixed-signal neuron (proposed). All SC-NNs with stochastic bitstream length of 1. All ANNs are implemented as maximal parallel ANNs, i.e. each input is processed by a separate MAC operation.

	Method	Delay	Area	Power	#EGTs
Neuron	4-bit ANN	55.9ms	138.7mm <sup>2</sup>	3.292mW	992
	8-bit ANN	242.55ms	3174mm <sup>2</sup>	123mW	19873
	4-bit SC-NN	47ms	372mm <sup>2</sup>	51mW	2542
	8-bit SC-NN	191ms	3466mm <sup>2</sup>	202mW	21453
	<b>Proposed</b>	<b>108ms</b>	<b>23mm<sup>2</sup></b>	<b>1.12mW</b>	<b>144</b>

The input features were normalized to a range of  $[-1, 1]$  for all datasets. All neural networks were trained for 200 epochs with the classical mean-squared error loss function. The topology of the networks was fixed at  $\#input \times 3 \times \#output$ . Training/testing was performed using a random 67%/33% split.

As can be obtained from Table 4.7, the trained stochastic computing ANNs exceeded the accuracy of the random guess (baseline) substantially for 9 of the 13 datasets. Furthermore, the stochastic computing ANNs have only small variations on the inference result, smaller than 3%. Also, the proposed ANN achieves similar accuracy compared to a 4-bit digital ANN (deterministic) for a few datasets. One reason for the different accuracies between tasks is their varying difficulty. Second, some of them also do not allow for a 100% correct classification based on the chosen input features.

It is important to mention, that further improvements are expected after a hyper-parameter optimization performed individually for each dataset to find the optimal topology (e.g., with more hidden nodes and layers). But in summary, it can be validated that the proposed stochastic SC-NN is capable of performing classification task, with less than 13% inference



**Figure 4.13.:** Comparison between digital ANN, digital SC-NN and mixed signal SC-NN (proposed) for topology: 9-3-2 (all NNs based on EGTs). All SC-NNs with stochastic bitstream length of 1. Average accuracy across all datasets is also provided. All NNs are implemented as maximal parallel NNs, i.e., each input is processed by a separate MAC operation.

accuracy loss compared to the unconstrained ANN (hardware-agnostic) for a majority of the benchmark datasets.

#### 4.2.5. Summary

In this chapter, new designs in the stochastic computing domain were explored, which enable a low-complexity neural network architecture, which is feasible to be fabricated by PE technology. Post-fabrication tuning, which is a feature of an additive printing process, is leveraged to implement stochastic computing components with very low hardware footprint, beyond the capabilities of conventional digital realizations. As a result, the area/power consumption of the proposed mixed-signal SC-based neural network is only 25%/35% of a 4-bit digital implementation and 10%/3% of a conventional 4-bit SC-based ANN. Moreover, the architecture enables arbitrary stochastic bit stream lengths, which can lead to a reduction in variance of classification estimates at the expense of higher ANN inference delay.

Another advantage of using SC in general is noise immunity, which means that bit-flipping has only marginal impact on the computation result. Although the output of SC operations is naturally stochastic, the accuracy of computations can be increased by enlargement of the random bit streams, also called progressive precision. In summary, the low-complex mixed-signal SC-NN is particularly interesting for direct sensor readout and conditioning in PE, for applications where silicon-based technology is not an option. This is the first time, a stochastic computing neural network was implemented and evaluated in PE.



## 4.2. Inkjet-Printed Stochastic Computing Neural Networks

**Table 4.7.:** Comparison of ANN inference results for hardware-agnostic ANN (unconstrained weights, true ReLU), a PE specific ANN (weights bounded, stochastic adder, bi-polar ReLU), deterministic 4-bit and 8-bit ANNs (fully digital), and the stochastic computing ANNs (random bit stream length of 1024) - both 4-bit and 8-bit digital implementations and mixed-signal SC (proposed) implementation. Both average and 1-sigma confidence interval ( $\pm$ ) are included. Also, random guess is provided as a baseline. For the hardware-agnostic and PE-specific ANNs, also inference results on the train/test-split are shown.

Dataset	Topology	Hardware -Agnostic		PE Specific		Deterministic		Stochastic			Baseline
		Train	Test	Train	Test	4-bit	8-bit	4-bit	8-bit	Proposed	Random Guess
Acute Inflammation	6-3-2	1.0	1.0	1.0	1.0	1.0	1.0	0.920 $\pm$ 0.029	0.915 $\pm$ 0.028	0.918 $\pm$ 0.03	0.475
Balance Scale	4-3-3	0.914	0.903	0.900	0.874	0.852	0.914	0.874 $\pm$ 0.008	0.872 $\pm$ 0.008	0.874 $\pm$ 0.008	0.440
Breast Cancer Wisconsin	9-3-2	0.970	0.965	0.955	0.952	0.967	0.970	0.714 $\pm$ 0.009	0.714 $\pm$ 0.008	0.714 $\pm$ 0.008	0.667
Energy efficiency (y1)	8-3-3	0.889	0.882	0.870	0.850	0.881	0.889	0.793 $\pm$ 0.013	0.794 $\pm$ 0.013	0.791 $\pm$ 0.014	0.433
Energy efficiency (y2)	8-3-3	0.909	0.890	0.883	0.862	0.885	0.905	0.748 $\pm$ 0.013	0.750 $\pm$ 0.012	0.749 $\pm$ 0.015	0.465
Iris	4-3-3	0.990	1.0	0.910	0.860	0.96	0.99	0.684 $\pm$ 0.01	0.6837 $\pm$ 0.011	0.684 $\pm$ 0.009	0.280
Mammographic Mass	5-3-2	0.838	0.833	0.824	0.780	0.812	0.835	0.756 $\pm$ 0.009	0.755 $\pm$ 0.009	0.755 $\pm$ 0.009	0.550
Seeds	7-3-3	0.979	0.986	0.921	0.914	0.942	0.971	0.867 $\pm$ 0.018	0.869 $\pm$ 0.017	0.868 $\pm$ 0.018	0.271
Tic-Tac-Toe Endgame	9-3-2	0.991	0.975	0.989	0.978	0.808	0.990	0.668 $\pm$ 0.004	0.668 $\pm$ 0.004	0.668 $\pm$ 0.004	0.640
Vertebral Column (2 cl.)	6-3-2	0.855	0.903	0.768	0.786	0.812	0.850	0.559 $\pm$ 0.022	0.559 $\pm$ 0.022	0.555 $\pm$ 0.024	0.690
Vertebral Column (3 cl.)	6-3-3	0.768	0.806	0.647	0.670	0.469	0.768	0.297 $\pm$ 0.026	0.290 $\pm$ 0.023	0.297 $\pm$ 0.022	0.515
Cardio	21-3-3	0.839	0.826	0.784	0.766	0.802	0.826	0.784 $\pm$ 0.0006	0.784 $\pm$ 0.0003	0.784 $\pm$ 0.0005	0.766
Pendigits	16-3-10	0.518	0.520	0.501	0.509	0.441	0.526	0.216 $\pm$ 0.004	0.218 $\pm$ 0.002	0.219 $\pm$ 0.004	0.099



## 5. Inkjet-Printed Analog Circuits

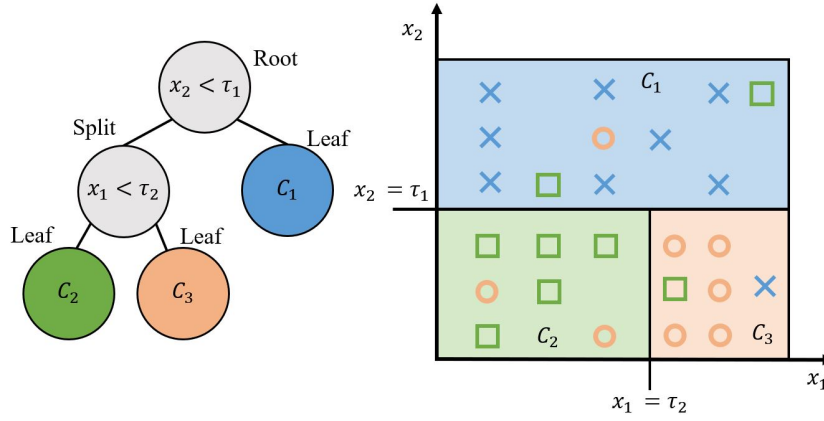
In this chapter, similar to the neuromorphic computing circuits presented in Chapter 4, the design space is further explored for low-complexity electronic components, which are expected to play a major role in future printed applications. From a study on machine learning classifiers it was revealed, that binary decision tree classifiers - which are besides artificial neural networks another popular tool for machine learning problems - provide a good balance in terms of hardware costs and inference accuracy [33]. Due to this, in this chapter several printed decision tree designs were evaluated, using conventional (digital), bespoke and analog computing paradigms. In the second section of this chapter, an implementation of a printed read-only-memory (ROM) from the analog domain was studied, which is in particularly interesting to reduce area utilization of memory devices. The proposed one-time programmable ROM element can be used furthermore as a replacement of combinational logic for lowering hardware costs [33].

### 5.1. Inkjet-Printed Decision Tree

Decision trees [83], as tree-based regression and classification models form the basis for many powerful machine learning methods such as Random Forests [84]. With decision trees, a set of input data points  $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$  which consist of pairs of feature vectors  $\mathbf{x} \in \mathbb{R}^n$  and class labels  $y \in \mathbb{Y} = \{c_1, c_2, \dots\}$ , are partitioned by a set of split-functions  $f_j(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}, j = 0, \dots, m$ . By iterative application of the split functions, the feature space is divided into a set of hypercubes, to which a class label is assigned [85]. Depending on the outcome of the evaluation of the first split function, a subsequent split function is chosen, and this is repeated for a fixed number of steps, depending on the tree depth.

For a binary decision tree, each split-function divides the feature space into two distinct regions. A binary decision tree with depth of 2 and the associated feature space partition is depicted in Figure 5.1. Starting from the root node, the tree is traversed from top to bottom through split nodes. Based on the decision at each node (root or split nodes), either the left or right child node is selected and a new split function is invoked. This procedure is repeated until the leaf nodes are reached, which then define the classification outcome. In the illustrative example of Figure 5.1, the feature space is partitioned by the decision tree into three distinct regions and each region corresponds to class labels which are  $C_1, C_2$  and  $C_3$ .

Decision trees constructed by this procedure are simple to understand and provide transparent predictions (white box). Moreover, the capability of strong feature selection is in particular beneficial for printed sensor fusion applications, where only the most important sensor signals are relevant and chosen to be processed by the decision tree.



**Figure 5.1.:** *Left:* The illustration of a binary decision tree with a depth of 2. *Right:* Partitioning of the feature space based on the decisions of the illustrative example on the left. Data points are also depicted and labeled according to their related class (indicated by their color).

### 5.1.1. Conventional Digital Binary Decision Trees

As any decision tree classifier can be realized by binary trees, where each node has at maximum two child nodes, the focus is in the following on the implementation of binary decision trees without compromising the generality. The essential computation required for performing inference in a decision tree is the decision operation, which can be implemented by a digital comparator. As the decision tree circuit designs were extracted from Boolean logic functions, they are termed in the following as conventional decision trees. Using the conventional design approach, the decision tree can be either implemented by a serial tree, or parallel decision tree design.

For the serial decision tree, only one comparator is deployed and the decision operations are performed sequentially. In addition to the comparator also memory devices are required which store the threshold values  $\tau$  and the class labels  $C_i$ . The outcome of the decisions are stored in an additional shift register [33], and the inference results are obtained at least after  $N$  steps, where  $N$  is the depth of the tree.

Due to the time-multiplexed computation of the serial digital decision tree, inference delay is high and increases with the depth of the decision tree. In contrast, a fully parallel digital decision tree performs all decisions simultaneously with similar inference delay, irrespective of the depth of the tree. The reduction in inference delay is however at the expense of hardware overhead, as multiple comparators are required, one for each node in the decision tree.

A quantitative comparison between digital serial and parallel trees is provided in Table 5.1. The pertinent parameters of the hardware implementations were obtained from synthesis results using the EGT standard cell library [63]. The threshold value and input features were decoded with 8-bit precision. As can be derived from Table 5.1, the serial digital decision tree (DT) has very high inference delay (197.3ms for DT-8), while the parallel decision tree requires only 59ms. On the other hand, the serial decision tree has area overhead of  $400\text{mm}^2$  for a depth-8 decision tree, compared to  $21\,300\text{mm}^2$  for the parallel decision tree. Nevertheless, both hardware implementations are infeasible to be fabricated in PE due to the large area utilization and power consumption overheads. Thus, in the following, decision tree designs are presented, which have substantially lower hardware footprints.

**Table 5.1.:** Conventional digital serial and parallel decision trees with depth of 1 (DT-1), 2 (DT-2), 4 (DT-4) and 8 (DT-8) [33]

Trees	Serial				Parallel			
	Delay	Area	Power	Gates	Delay	Area	Power	Gates
DT-1	20.14ms	20.05mm <sup>2</sup>	1.65mW	59	29ms	155mm <sup>2</sup>	13.1mW	257
DT-2	48.48ms	40mm <sup>2</sup>	5.09mW	115	37ms	420mm <sup>2</sup>	30.6mW	790
DT-4	94.96ms	191.6mm <sup>2</sup>	28.35mW	626	50ms	2000mm <sup>2</sup>	118mW	4.1k
DT-8	197.3ms	400mm <sup>2</sup>	71.71mW	638	59ms	21 300mm <sup>2</sup>	1027mW	49k

### 5.1.2. Bespoke Digital Binary Decision Trees

Due to the low fabrication costs, inkjet-printed bespoke designs can be generated and manufactured on-the-fly during on-site fabrication depending on the particular use case, which reduces the pressure for having reusable printed hardware. This allows for the realization of machine learning classifiers which are tailored to the target application. In contrast to conventional decision trees, the threshold values which are derived from the training cycle, are hardwired into the hardware design. While this makes the decision tree-based inference engine less flexible and trained for only one specific dataset and use case, substantial reductions in delay, area and power are achieved.

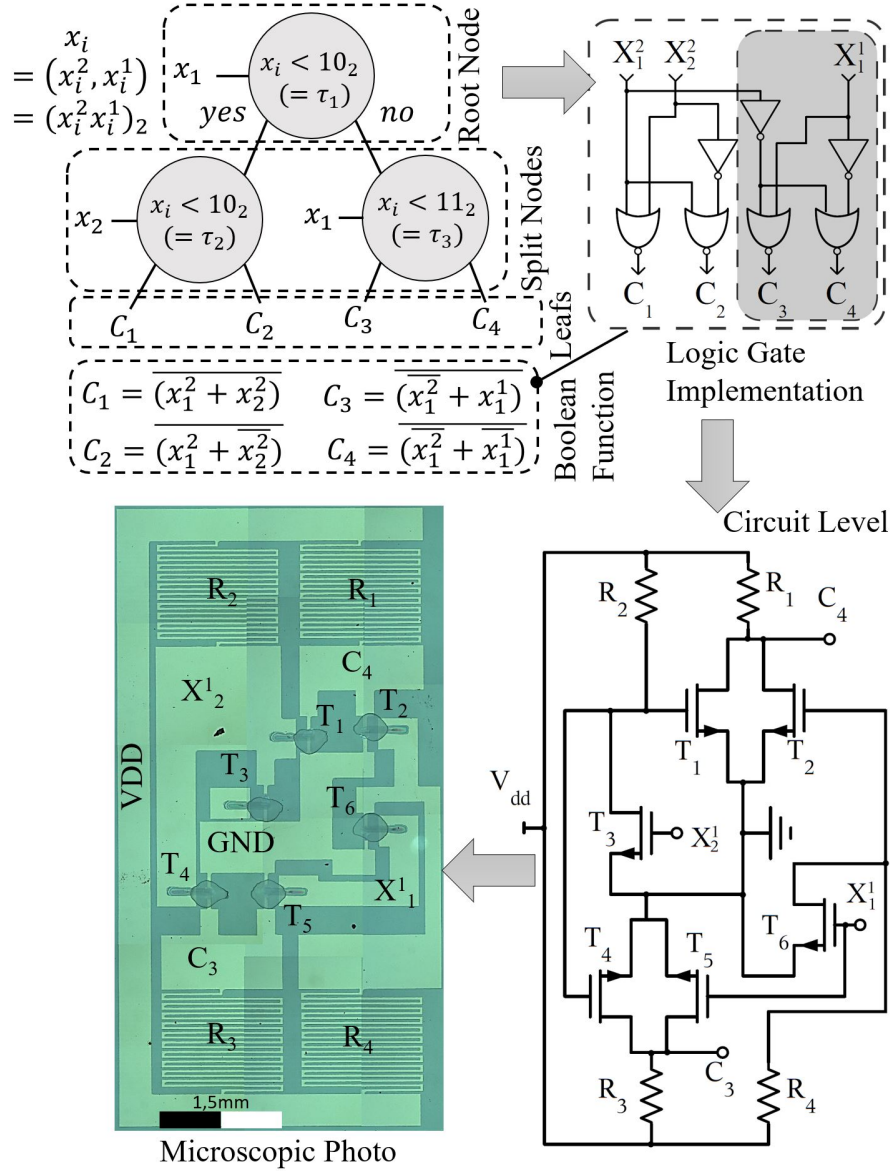
The design flow for the fabrication of a bespoke decision tree circuit is illustrated in Figure 5.2, which depicts a balanced binary decision tree with depth of 2 and fixed threshold values. The input feature vector  $(x_1, x_2)$  is two-dimensional and each vector component has a 2-bit precision ( $x_i = (x_i^2, x_i^1)$ ). In the root node the comparison is performed between the first feature vector component ( $x_1$ ) and the binary threshold value ( $10_2$  (binary) =  $2_{10}$  = 2 (decimal)). As the threshold values are fixed, a Boolean function for the class labels  $C_i$  can be derived, which is only dependent on the input feature vector  $C_i = f(x_1, x_2)$ . Based on the Boolean function of the class labels, a logic gate implementation is extracted, which is preferably represented by inverters, NANDs and NORs, i.e. components which can directly be mapped to the circuit level. The right part of Figure 5.2 provides how the logic gate design is converted to the circuit level, which is then in a final step transformed into the layout of the circuit (microscopic photo). In general, any kind of bespoke decision tree design generation can easily be automated by following the proposed design flow described in Figure 5.2.

Similar to the conventional digital decision tree, high-level synthesis tools were deployed to implement larger designs with different tree depth. Again, a serial bespoke tree as well as a parallel bespoke tree were synthesized. In contrast to the conventional decision trees, the bespoke parallel decision tree performs better in all relevant parameters than the serial version [33]. Delay is  $9.51\times$  lower, area  $10.3\times$  less and power consumption reduced by  $28.8\times$  compared to the bespoke serial implementation. In comparison to the conventional parallel decision tree, improvements are: delay  $3.9\times$  lower, area  $48.9$  lower and power consumption reduced by  $75\times$ .

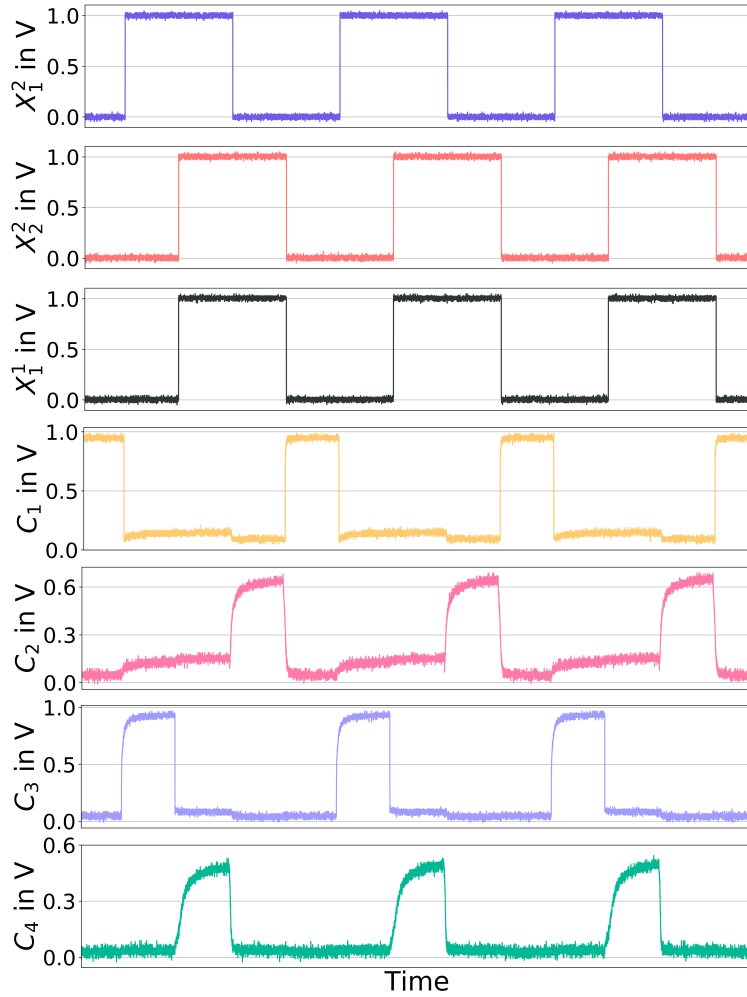
The bespoke decision tree design was validated by a printed hardware prototype depicted in Figure 5.2. The bespoke decision tree was fabricated according to Section 2.4. As all decision tree nodes were converted into a logic gate implementation, the resulting design is a maximally parallel decision tree design. The correct circuit functionality was confirmed by transient measurements provided in Figure 5.3. Due to the hardwiring of the decision tree thresholds,

## 5. Inkjet-Printed Analog Circuits

from the four input bits, only three are relevant for determining the correct class label. The relevant bit positions are:  $x_1^2, x_1^1$  (most significant bit (MSB) and least significant bit (LSB) of feature vector component  $x_1$ ) and  $x_2^2$  (MSB of feature vector component  $x_2$ ). All possible input combinations were applied to the circuit, and the output voltages were measured at the output class label pins:  $C_1, C_2, C_3, C_4$ . As can be derived from Figure 5.3, only one class label  $C_i$  is activated at a time. This is in accordance to the Boolean function description in Figure 5.2 and the decision tree is operating correctly.



**Figure 5.2.:** *Top:* Design flow of the proposed hardware prototype of a 2-bit balanced and depth-of-2 bespoke digital decision tree. Top-level design of the hardwired decision tree is converted into logic level representation.  $x_i$  denotes the input variable, while  $x_i^2/x_i^1$  (or  $X_i^2/X_i^1$ ) represent the MSB/LSB. *Bottom:* Logic-level design is converted into transistor level circuit description. Finally, layout-extraction and microscopic photo of the fabricated decision tree is extracted.



**Figure 5.3.:** Measured waveforms of the printed bespoke digital decision tree with 2-bit precision

### 5.1.3. Analog Binary Decision Trees

While bespoke decision tree architectures result in considerable area and power improvements over conventional architectures, the area usage and number of transistors is still high. Another effective method to reduce transistor count is by substitution of the multi-bit comparators by small analog circuits, which require only a few transistors. In the following, an analog decision tree architecture is provided, which is besides the previously discussed neuromorphic computing systems (Chapter 4) another example of mapping machine learning classifiers efficiently to printed hardware.

To build an analog decision tree, the most expensive operation during inference is the if-else-statement of the form:  $x_k \leq \tau_j$ , where  $\tau_j$  is a pre-defined threshold obtained from the learning phase and  $x_k$  the input variable. Instead of deployment of multi-bit comparators, in an analog design this binary comparison can be realized by a back-to-back inverter, which has a transistor in the pull-up network of one of the inverters, as can be obtained from Figure 5.4a. Furthermore, the input signals  $x_k$  are encoded as continuous-valued voltage signals, normalized and bounded according to the feasible supply voltage ranges in EGT technology:  $[0V, 2V]$ . In the analog decision tree nodes, the input signals (e.g.  $x_1, x_2$ ) are applied to the transistors

## 5. Inkjet-Printed Analog Circuits

(e.g.,  $T_1, T_4, T_7$ ) and converted into a transistor drain-source resistance proportional to their voltage level. Depending on the input signal voltage level, the transistor resistance varies over the resistance range ( $[R_{on}, R_{off}]$ ), where  $R_{on}$  is the on-state resistance ( $x_i = 2V$ ) and  $R_{off}$  the off-state resistance ( $x_i = 0V$ ) of the transistor. If the transistor resistance (e.g. of  $T_1$ ) is lower than the pull-up resistor of the opposite inverter (e.g.  $R_1$ ), the adjoint decision tree output node (e.g.  $S_1$ ) is pulled up to VDD, while the output node of the opposite inverter is pulled down to ground. If the transistor resistance is higher than the pull-up resistor, the opposite case occurs. In either case, the bi-stable back-to-back inverter converges to a stable state, where the output nodes are complementary ('1'/'0' or '0'/'1'). This is how the decision is implemented, and by passing the output signals (e.g.,  $S_1$  or  $S_2$ ) to the child nodes, also the selection operation is realized. This guarantees, that only one child node is selected per tree level and thus only one leaf at a time is selected after the inference completes.

For the analog decision node, the threshold  $\tau_j$  is encoded as a resistor, by using the following mapping function:

$$R_j = \frac{\tau_j - \tau_j^{min}}{\tau_j^{max} - \tau_j^{min}} \cdot (R_{max} - R_{min}) + R_{min}, \quad (5.1)$$

where  $R_{min}, R_{max}$  represent the feasible range of printed resistor values, and  $\tau_j^{min}, \tau_j^{max}$  are the threshold value bounds determined during the decision tree training phase.

The threshold resistors in EGT technology can be fabricated by inkjet-printed conductive materials, such as PEDOT:PSS. Their resistance can be adjusted by changing the lateral resistor geometries as well as the printed film thickness. An interesting side-effect of the proposed analog decision tree architecture is that by selection and deselection of portions of the decision tree circuit the overall power consumption is reduced similar to power-gated circuits.

For the construction of larger decision tree designs, split nodes are repeatedly added to the last layer until the desired tree depth is reached. It is important to note, that due to the insertion of selector transistors in each split node (e.g.,  $T_1^S, T_2^S$ ), the voltage swing in the subsequent child nodes deteriorates. However, for larger decision trees, the voltage signals can be restored by the insertion of buffer elements (e.g., inverter-based) placed before the input of the selector transistors.

Compared to the digital bespoke maximal parallel tree, the area usage is reduced exorbitantly by  $436\times$ . Power consumption is  $26\times$  lower, while the delay was slightly increased by  $1.6\times$ .

In order to demonstrate the feasibility of the analog decision tree classifier design, an analog 2-level decision tree was designed in EGT-technology. The layout was already presented in Figure 5.4a, and it consists of one root node with two split nodes and four leaves, which results in 11 EGTs and 3 printed resistors. The PEDOT:PSS-based resistors are printed to three locations at the interface to VDD and  $S_2$  ( $R_1$ ), source of  $T_1^S$  and  $C_1$  ( $R_2$ ) and source of  $T_2^S$  and  $C_4$  ( $R_3$ ). The fabrication process was according to Section 2.4. Similar to the presented printed LUT (Section 3.2), crossovers were required at two locations and implemented by an isolation layer (Dimethylsulfoxide (DMSO) and Polycarbonate (PC)) in combination with a conductive layer (PEDOT:PSS) on top. The microscopic photo is depicted in Figure 5.4b.

To validate proper operation of the hardware prototype, transient measurements were performed. First the functionality of the root node was tested by pulsing the input signal  $x_1$



between 0V and 2V and recording the output node voltages  $S_1$  and  $S_2$ . As expected,  $S_1/S_2$  are in state ‘1’/‘0’ when  $x_1$  is at logical ‘1’ and in state 0’/‘1’ when  $x_1$  is at logical ‘0’ (see Figure 5.4b right). For the split nodes, all four input combinations were tested. For split node B (“Split B”), both output voltages  $C_3/C_4$  are pulled down to 0V when the split node is unselected ( $x_1$  is high). While the leaves  $C_3/C_4$  are pulled up or down, according to the input signal  $x_2$ , when the split node is selected ( $x_1$  is low). As can be seen, the worst case output signals of the split node can be distinguishable by 405mV, and hence the prototype was functioning correctly.

#### 5.1.4. Training

Training of the proposed printed decision trees is similar to a hardware-agnostic decision tree design. After obtaining the split function and threshold values from the hardware-agnostic training phase, they are rounded to the next binary value (for digital decision trees only, dependent on the bit-precision) or by converting them into resistances according to Equation (5.1) (analog decision tree).

In general, finding the optimal tree structure is usually computationally infeasible, due to the large number of combinations of possible solutions [58]. The determination of the structure of the decision tree is an iterative process using a greedy algorithm. An initial tree is built which consists of only one root node. Now by an exhaustive search, many different choices of feature selections (input variable  $x_i$ ) and threshold values for the root node are explored, and the selection is made according to a measure, which is least-mean-square error for regression problems, and cross-entropy or “Gini index” for classification problems [58]. The cross entropy is defined as:

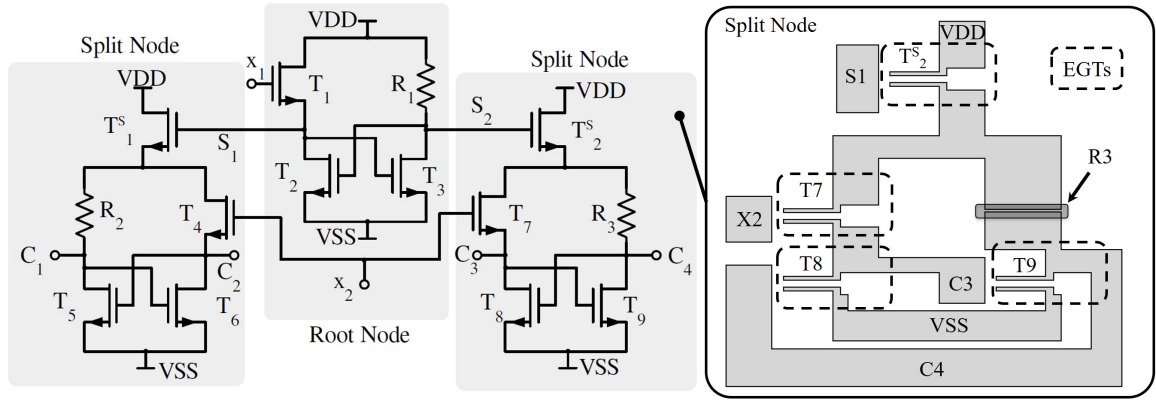
$$Q_r(T) = \sum_{k=1}^K p_{rk} \ln(p_{rk}), \quad (5.2)$$

while the Gini index is computed by:

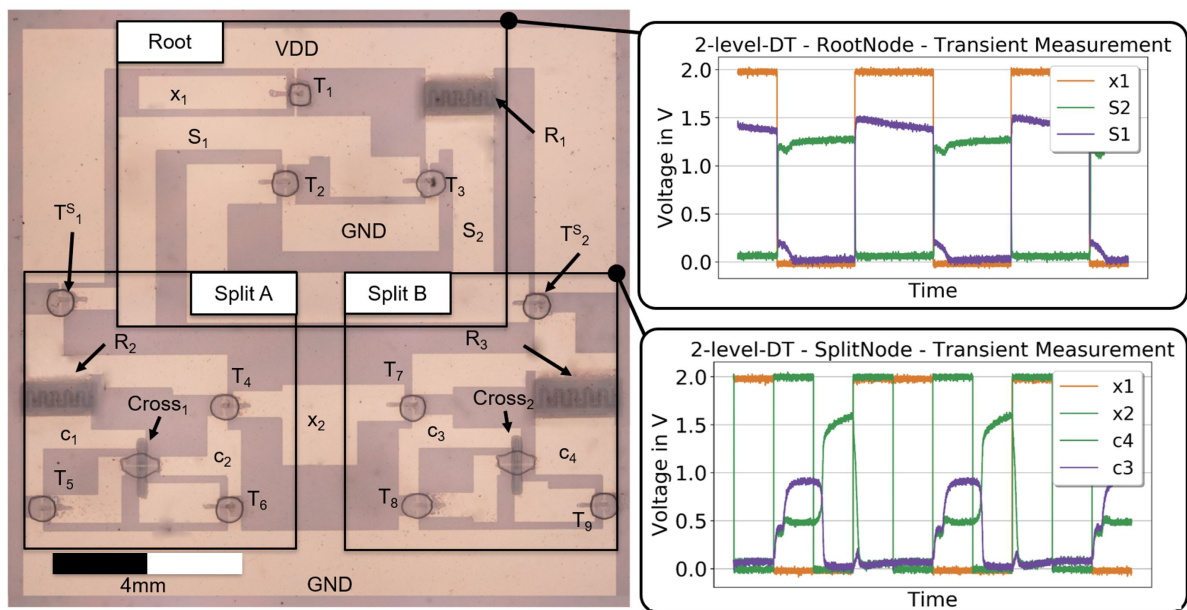
$$Q_r(T) = \sum_{k=1}^K p_{rk} * (1 - p_{rk}), \quad (5.3)$$

where  $Q_r(T)$  is a term to be minimized during training, T is the tree topology, K is the number of class labels,  $r$  denotes the region and  $p_{rk}$  is the proportion of data points in region  $r$  which are label as  $k$ .  $\sum_r Q_r(T)$  is the overall contribution to the residual and the objective to be minimized during training. As can be obtained by Equation (5.2) and (5.3), splits are favored where  $p_{rk}$  is either 1 or 0 for which  $Q_r(T)$  vanishes. This is also termed as “purity” of a distribution.

As there exist many possibilities for thresholds and feature selections, region bounds are usually axis-aligned [58] and can be defined by the split function:  $f_j(\mathbf{x}) = x_k - \tau_j$ . After the boundaries are determined by optimizing Equation (5.2) or (5.3), child nodes are added to the root node and the procedure is repeated until a termination criterion is fulfilled. Most commonly, large tree sizes are first grown and then pruned until having an optimal trade-off between inference accuracy and tree size.



(a) Schematic and Layout of 2-level analog decision tree



(b) Microscopic photo and measurements of 2-level analog decision tree hardware prototype

**Figure 5.4.:** a): Schematic and layout of a 2-level analog decision tree with one root and two split nodes. Each node is implemented by a back-to-back inverter with a transistor in the pull-up network of one inverter. The split nodes have additionally a selector transistor which connects/disconnects the inverters to/from VDD, depending on whether the node is selected ( $S_j$  is high) or unselected ( $S_j$  is low). b): a digital post-processed image from multiple microscopic photos of the printed 2-level analog decision tree is provided with the corresponding transient measurements on the right.

For the prediction performed by the trained decision tree classifier, the regions are labelled according to the most frequent data points in the region, which define then the classification outcome. During inference, it is observed in which region  $r$  the input vector  $x$  falls by applying all split functions  $f_j$  based on the individual decision made by traversing the tree from root node to leaf node. For the experiments, the decision trees were trained using the scikit-learn library

**Table 5.2.:** Accuracy and computation requirements for different classification algorithms and decision trees with depth of 1 (DT-1), 2 (DT-2), 4 (DT-4) and 8 (DT-8) [33]. Models trained by scikit-learn [86]. A: Accuracy on test data, #C: Number of comparisons

	DT-1		DT-2		DT-4		DT-8	
	A	#C	A	#C	A	#C	A	#C
Arrhythmia	0.56	1	0.58	3	0.62	12	0.60	46
Cardio	0.79	1	0.79	3	0.87	13	0.94	47
GasID	0.37	1	0.49	3	0.67	14	0.90	97
HAR	0.57	1	0.82	2	0.90	4	0.99	4
Pendigits	0.19	1	0.32	3	0.66	15	0.92	122
RedWine	0.44	1	0.47	3	0.53	14	0.51	112
WhiteWine	0.45	1	0.48	3	0.51	15	0.54	160

[86]. The inference results after training are provided in Table 5.2 and several observations can be made. Firstly, for some datasets the trained decision trees do not perform well (RedWine, WhiteWine), as even the DT-8 cannot improve the accuracy beyond 54%. On the other side, for a few datasets (Cardio, HAR) even the smaller decision trees (DT-2, DT-4) provide acceptable inference accuracy. Finally, it can be seen that the required computational resources increase substantially (97 comparisons) for one dataset (GasID) where high accuracy can only be obtained by using the DT-8 classifier. This further motivates the implementation of the low-complex analog decision trees introduced in this section.

### 5.1.5. Summary

In this chapter, different computing paradigms were explored based on printed decision trees, which are deployed as inference engines for machine learning classifications. Such machine learning classifiers provide a good balance between hardware cost and accuracy [33]. By starting from conventional digital decision tree implementations, it was demonstrated how bespoke designs can improve the high area and power consumption overheads of printed decision tree hardware. Although complexity of bespoke hardware is decreased, still transistor count is very high for printed applications. Thus, analog decision tree designs were proposed, which decrease the hardware footprint even further.

An evaluation on hardware costs of printed decision trees was carried out on popular classification benchmark data sets for all three design points. The bespoke parallel decision tree design improves area usage by 25%, power consumption by  $6.8\times$ , and delay by  $8\times$  compared to a conventional serial decision tree design. However, the smallest hardware footprint is achieved by the analog decision tree design, which has  $436\times$  smaller area usage and  $26\times$  reduced power consumption compared to the bespoke parallel decision tree, but at the expense of an increase in delay by 60%.

From the conducted experiments, it can be concluded that digital parallel bespoke decision trees perform the best regarding inference delay, while analog decision trees offer more efficient area usage and power consumption. Another advantage of analog decision trees is the capability of direct sensor readout, e.g. analog sensor data can directly be interfaced and processed without the deployment of expensive ADCs.

## 5. *Inkjet-Printed Analog Circuits*

To the best of my knowledge, this is the first time that benefits of bespoke and analog decision tree classifiers were quantified in PE. Also, this is the first time that digital and analog decision trees were prototyped in PE. The presented tree-based classifier concepts can be further explored in future by considering more complex models such as random forests or gradient-boosted trees.

## 5.2. Inkjet-Printed Analog Read-Only Memory

Read-Only-Memories (ROMs) are commonly deployed for digital filters, microprocessors [87], computers and other electronic devices. As a non-volatile memory device, stored data is not changed during the system lifetime. Also in the context of PE, such devices would be the preferred choice to implement low-cost and high-density memory arrays.

Traditional architectures are based on the ROM matrix with additional circuitry, the decoder logic. Due to the stringent requirements on chip area in PE, ROM designs with high information density are favored. A conventional technique to increase the memory density is to shrink the memory cells in the ROM matrix by using multiple states of information for a single cell [88]. While usually one 1 bit of information is stored per cell, different methods exist to obtain multi-valued data, such as transistor threshold voltage variation or transistor channel size variation [88].

For EGT-based ROMs, process-induced variations caused by non-determinism in droplet printing or humidity effects have a non-negligible impact on the EGT performance, and as a result the aforementioned techniques are inappropriate. A more promising approach is to use a crossbar architecture for the ROM matrix, where data is hardwired by printing different shaped resistors at the crossbar interconnects, which are less susceptible to process variations or environmental effects than EGTs. The shape of printed resistors can be easily controlled, and as it is shown in the following, 4 different resistance states could be encoded, which leads to 2-bit information per memory cell.

Also the hardware costs of such a resistor-based ROM are comparable to combinational logic in EGT-technology. For example, a 1-bit EGT ROM has an area of  $0.05\text{mm}^2$ , while an EGT-based inverter has an area of  $0.22\text{mm}^2$  [63], and a power consumption of  $3.13\mu\text{W}$  and  $9.6\mu\text{W}$ , respectively. Also in terms of delay, a ROM cell in EGT-technology is within  $1.5\times$  of an inverter cell [63]. On the other hand, the pertinent parameters between a ROM cell and an inverter in silicon are out of proportion. For instance the ROM cell delay in silicon is  $900\times$  higher than an inverter cell [33]. In addition, the power consumption is  $\sim 1200\times$  higher [33].

To demonstrate the feasibility of a ROM element in EGT-technology, a  $4\times 1$  one-time programmable ROM element was fabricated and characterized. The printed ROM hardware prototype consists of one row and four columns, as can be obtained from the schematic in Figure 5.5a. Each of the four columns of the ROM are accessed by transistor-based decoder logic, implemented by the transistors  $T_1 - T_4$ . The data is stored in the resistive crossbar by printing multi-valued resistors  $R_1 - R_4$  at the crossbar interconnects. During a read operation, one of the four transistors is turned on, and the output voltage  $V_{out}$  across the sensing resistor  $R_{sense}$  is measured. The output voltage depends on the ratio of the printed resistor  $R_i$  and the fixed sensing resistor  $R_{sense}$ , which form a voltage divider. The output voltage can be computed by (assuming no voltage drop at the decoder logic):

$$V_{out} = \frac{R_{sense}}{R_{sense} + R_i} \cdot VDD \quad (5.4)$$

By varying the geometry of the printed resistors  $R_1 - R_4$ , different resistance states are obtained and multiple values are encoded in the memory cell. As it also can be obtained from the chosen widths of each individual printed resistor (length constant) in Figure 5.5b, the resistances were set to:  $R_1 = 2 \times R_{sense}$ ,  $R_2 = \infty$  (not printed),  $R_3 = R_{sense}/2$  and  $R_4 \sim 0\Omega$

(maximum resistor area).

As a result, four different states lead to 2-bit of information per ROM cell, and in total 8-bit of information can be encoded for the  $4 \times 1$  ROM. The functionality of the 8-bit ROM is validated by the transient measurements in Figure 5.5c. In the measurement, only one of each column was selected at the same time, and the output voltage  $V_{out}$  was recorded. According to the four different resistance states, four different output voltages were measured: 0.5V ( $T_1$  activated by  $V_1$ , current sensed through  $R_1$ ), 0V ( $T_2$  activated by  $V_2$ , current sensed through  $R_2$ ), 0.75V ( $T_3$  activated by  $V_3$ , current sensed through  $R_3$ ) and 1V ( $T_4$  activated by  $V_4$ , current sensed through  $R_4$ ). This confirms that 2 bits of information could be encoded per cell. The measured delay of the prototyped ROM element was about 10ms with an area requirement of  $38\text{mm}^2$  and an average power consumption of  $39\mu\text{W}$ .

### 5.2.1. Summary

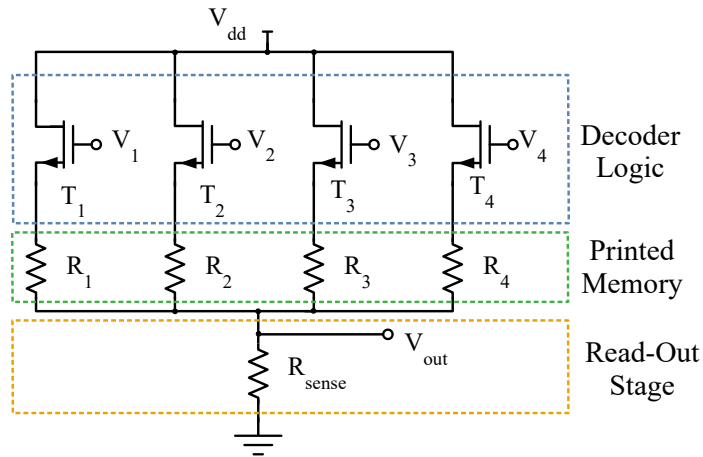
Regarding the presented prototyped 8-bit ROM, several optimizations can be performed. First of all, the memory size of the ROM can be up-scaled by adding additional rows and columns. However, the decoder logic also has to be increased, and effects such as sneak currents have to be considered (can be avoided by adding diodes to the crossbar rows). Due to the voltage divider-based read operation, resistors with higher resistances can be printed, e.g. by scaling both the resistors in the ROM matrix and the sensing resistor in the read-out stage by the same factor, without changing the output voltages. This fact can be derived from the following formula:

$$V_{out} = \frac{R_{sense}}{R_{sense} + R_i} \cdot VDD = \frac{R_{sense} \cdot s}{R_{sense} \cdot s + R_i \cdot s} \cdot VDD, \quad (5.5)$$

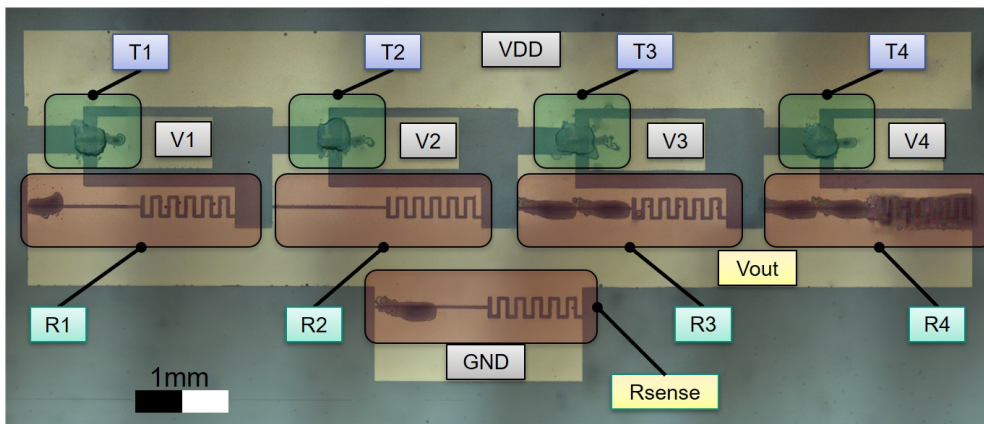
where  $s$  is the scaling factor. This improves area usage (larger resistors can have shorter widths) and power consumption (larger resistors decrease the currents through the ROM cell). Finally, more bits could be encoded by increasing the number of resistance states, however this is bounded by the limitation of the sensing margin.

It is also worth drawing a comparison here to the presented SR-Latch introduced in Section 3.1. In contrast to the latch, the prototyped analog 8-bit ROM element can store  $8 \times$  more information, with the same number of EGTs. While both presented hardware prototypes require 4 EGTs, the latch can store only 1 bit of information, while each of the four columns of the ROM store 2 bit of information. However, the increase in information density is at the expense of configurability. In contrast to the latch, the proposed ROM element is only one-time programmable. However, as it is anticipated that printed systems might be used for short-shell-life applications, one-time-programmability might not be a concern. By deployment of the 8-bit ROM element, analog circuits such as the presented neuromorphic systems (Chapter 4) or analog decision trees (Section 5.1) can directly interface the ROM without resorting to expensive ADCs.

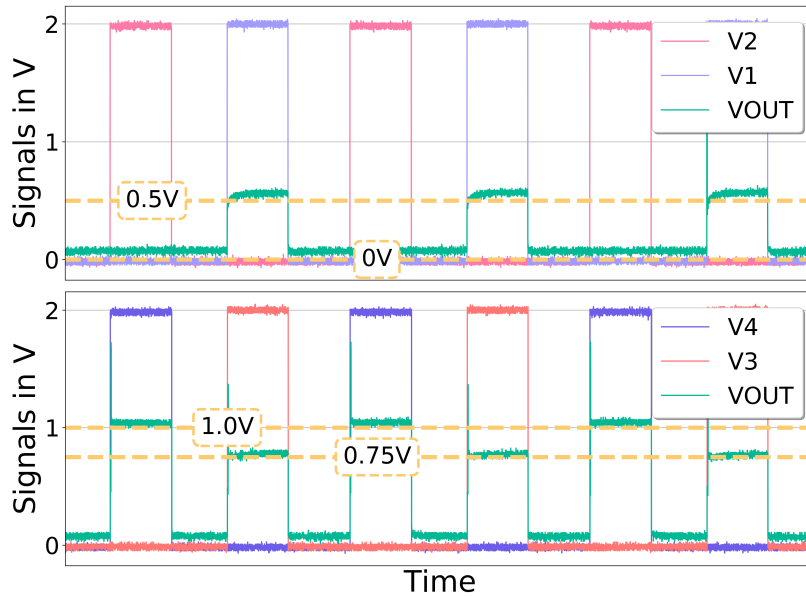
There are also key differences regarding silicon-based ROMs. While programmable ROMs in silicon are only economical when produced in large quantities, printed ROM have low cost per chip. For silicon-based ROMs, programming must occur early in the fabrication process [88], on the other hand, inkjet-printed ROMs are programmed during an on-site fabrication step and customized by the user, who defines the design and functionality of the ROM by a digital CAD software.



(a) Schematic of 8-bit ROM



(b) Digitally post-processed microscopic photo of printed 8-bit ROM hardware prototype



(c) Transient measurements of printed 8-bit ROM hardware prototype

**Figure 5.5.:** a/b): Schematic/Microscopic photo of a  $4 \times 1$  ROM. c): Transient measurements of the ROM hardware prototype.





## 6. Summary, Conclusion and Outlook

### 6.1. Summary

In this thesis, for the first time, several computing paradigms for the emerging inkjet-printed technology were evaluated in terms of design complexity, area usage, power consumption and performance. The presented hardware prototypes with their associated computing paradigms are summarized in Table 6.1. The explored design space consists of digital, analog, neuromorphic and stochastic computing. Moreover, for the majority of the hardware prototypes, bespoke circuits were derived by leveraging the customization capabilities of inkjet-printing technology.

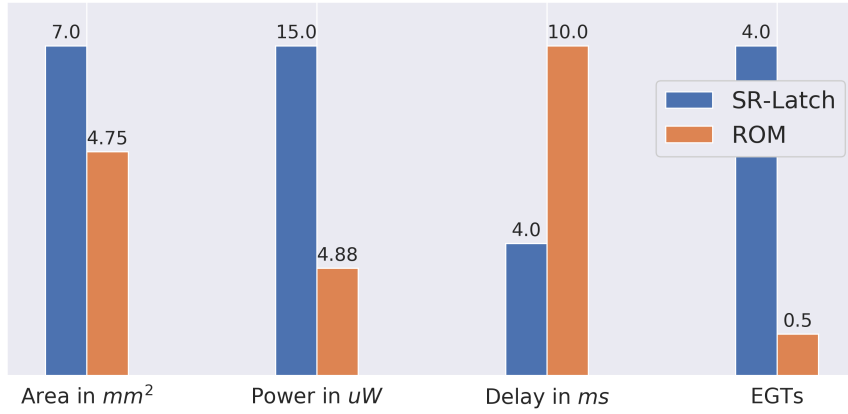
Two storage elements were designed, fabricated and characterized. The first memory device was a 1-bit SR-latch, whose design was derived from the digital computing domain and due to the absence of p-type transistors in EGT-technology, the latch - and also all other digital components in this work - were realized by resistor-transistor (RT) logic. On the other side, an 8-bit analog read-only-memory (ROM) was designed and fabricated, which is a one-time programmable memory element. As the stored information is encoded by the resistance value of printed and different-shaped resistors, 2-bit of information can be stored per memory cell. As depicted in Figure 6.1, reductions in area usage, power consumption and number of EGTs were achieved, however at the expense of higher device delay. Whenever a low-complex design is the main requirement, the proposed ROM concept is the preferred choice. However, as the ROM is a bespoke and one-time programmable circuit, only the proposed SR-latch enables sequential operations, e.g. required for the implementation of finite state machines.

An efficient implementation of digital combinational circuits in PE can be obtained by the proposed look-up table (LUT) design. By introducing a customization cycle, where printing of conductors to a circuit template is performed, any kind of Boolean logic function can be realized. The novel LUT design is in particular useful for additive printing technologies which offer an

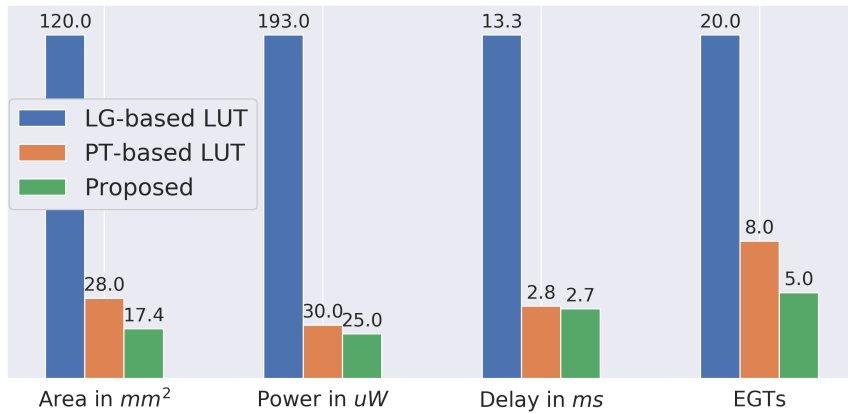
**Table 6.1.:** *Comparison of computing paradigms of the fabricated hardware prototypes in this work*

Circuit Designs	Computing Paradigms				
	Digital	Analog	Neuromorphic	Stochastic	Bespoke
SR-Latch	X				
LUT	X				X
Analog NN		X	X		X
SC-ANN (simulated)		X	X	X	X
Bespoke DT	X				X
Analog DT		X			X
ROM		X			X

## 6. Summary, Conclusion and Outlook



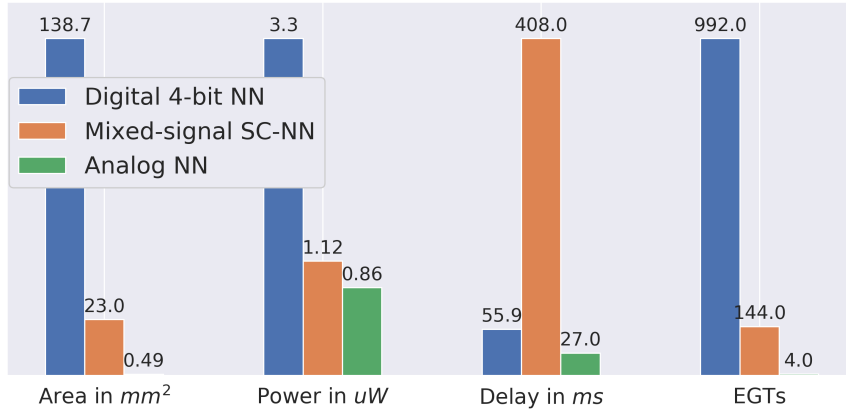
**Figure 6.1.:** Comparison of prototyped storage elements. Area usage, power consumption and number of EGTs are with respect to one stored bit of information. Thus, for one bit of information, only 0.5 EGTs are required for the ROM (and 4 EGTs for an 8 bit ROM (Section 5.2)).



**Figure 6.2.:** Comparison of printed LUT designs: conventional logic-gate-based (LG-based) LUTs, passing transistor-based (PT-based) LUTs and the proposed LUT.

on-site and on-demand fabrication process such as inkjet-printing, to enable low-complex LUT implementations. This leads to improvements in all pertinent parameters, as illustrated in Figure 6.2, and thus it is beneficial to use the presented approach for large-scale LUT-based architectures.

Two neuromorphic computing designs were analyzed in this thesis. The first concept implements neural algorithms by a printed analog neuron, which consists of a resistive synaptic crossbar architecture for realizing the MAC operation, and inverter-based circuits for implementation of negative weights and non-linear activation functions. The second proposed neuromorphic hardware design is based on stochastic computing (SC). Neuron functionality such as MAC operations are realized by very simple digital logic gates, and SC-based activation functions and stochastic signal generators are derived from novel analog circuits. Both approaches have benefits in terms of area usage, power consumption and number of required EGTs compared to a conventional digitally implemented artificial neural network (ANN), as can be obtained from Figure 6.3. However, the SC-based design for this experiment has large overheads in inference delay. For SC-NNs, inference delay depends on the chosen stochastic



**Figure 6.3.:** Comparison of a printed 3-input neuron using a digital 4-bit implementation, a mixed-signal stochastic-computing neuron (SC-NN) and an analog neuron. For the SC-NN, the bit length was set to 16 ( $=2^4$ ).

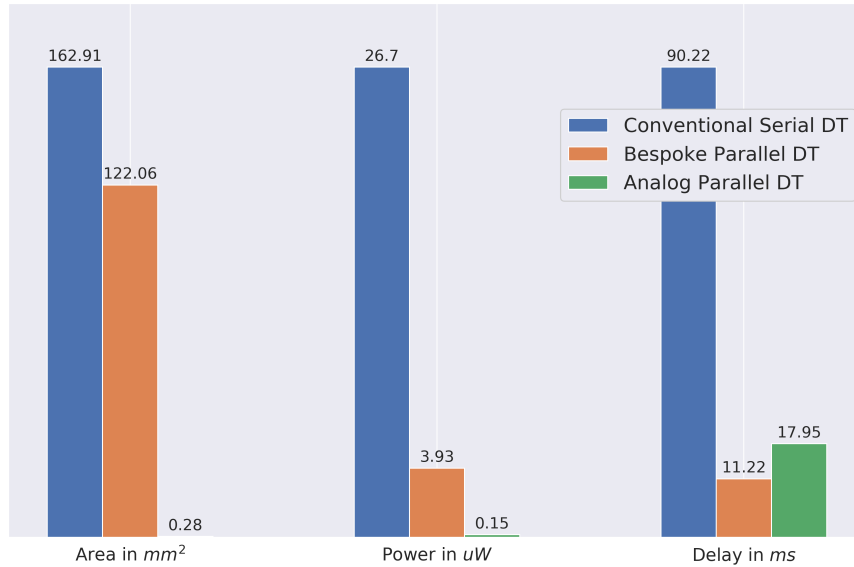
bit-stream length and cannot be compared directly with conventional ANNs. In Figure 6.3, the bit stream length for the SC-NN was set to 16. However, the proposed SC-NN architecture supports in general also shorter or larger stochastic bit stream lengths, which can be defined during run-time without changing the hardware design. This enables ANN inference with progressive precision (Section 4.2). Another benefit of SC-NNs is noise immunity to bit-flips, due to the inherently approximate nature of circuit operations in SC. Thus, it is anticipated that use cases exist where both proposed neuromorphic designs are favored over conventional digital ANNs.

Two decision tree designs (DT) were prototyped in this thesis. The first decision tree is based on digital computing but was designed as a bespoke circuit where the pre-trained decision tree thresholds were hardwired to reduce the power consumption and inference delay. This fact can be derived from Figure 6.4. By further exploring the analog computing design space, a circuit design was obtained, which substitutes the digital multi-bit decision tree comparators by very efficient analog designs, which require only very few EGTs. Compared to the bespoke decision tree, the analog design improves area usage and power consumption with a slight increase in inference delay. The presented design flow of the bespoke and analog decision tree designs allows for automated construction of large-scale decision tree architectures in the future.

## 6.2. Conclusions

EGT-based inkjet-printing is a promising candidate for future PE applications, as it enables point-of-use printed hardware due to its mask-less fabrication process. Despite previous achievements, it is still an open research question how circuit designs can be developed to implement machine learning classifiers in hardware, or how circuit customization can be leveraged in EGT-technology to reduce the usually high hardware costs. For this reason, this thesis explored circuit designs derived from several computing paradigms, such as digital, neuromorphic, analog and stochastic computing. From the conducted experiments and evaluations in this thesis, several conclusions can be drawn.

First of all, all kinds of digital circuits can be implemented in EGT-technology, by mapping



**Figure 6.4.:** Comparison of printed decision tree computing paradigms. Area, power and delay are obtained by averaging over decision tree implementations with a depth of 1, 2, 4 and 8 [33]. The conventional decision tree was implemented as a serial device to reduce the hardware footprint, thus only one comparator was implemented. On the other side, the bespoke and analog decision trees had to be implemented maximally parallel as the thresholds are hardwired and sequential operation is not an option.

existing CMOS-based designs to resistor-transistor logic for EGT-technology. In addition, the experiments in this thesis validate that also memory elements are manufacturable. Consequently, in combination with logic-gate-based combination logic, sequential computing can be performed e.g., for the implementation of digital finite state machines.

Furthermore, compared to other printing processes such as roll-to-roll and screen printing, inkjet-printing leads to much better area usage and power consumption due to the higher degree of customization. This can be concluded from the investigated bespoke circuit designs, deployed for the LUT, as well as the neuromorphic and analog circuits. Thus, inkjet-printing processes in PE are the preferred choice whenever high-volume fabrication is not required. Nevertheless, inkjet-printing can be deployed in conjunction with high-volume printing processes in the future to combine customization capabilities with high production throughput for split manufacturing.

Moreover, neuromorphic computing systems can be implemented in PE to map neural algorithms to printed hardware. By using analog or stochastic-computing paradigms, substantial improvements in area usage and power consumption are achieved compared to conventional digital implementations, which widens the applicability of printed machine learning classifiers enabling near-sensor computing in the future.

Also printed decision trees can be efficiently implemented in PE by using bespoke or analog circuit designs. Especially analog implementations are particularly interesting as they allow for direct sensor interfacing circumventing expensive ADCs.

It can also be concluded that parallel computations in PE are preferred over serial computations, as implementations of flip flops or latches have high transistor count and thus large hardware footprint. On the other side, wire costs in PE are negligible [33].

This work also emphasizes the well-known fact that printed hardware cannot compete with silicon-based circuits. It is not expected that design points exist, where printed electronics surpass silicon-based CMOS systems in terms of delay, area usage or power consumption [33]. However, this might encourage the use of hybrid systems (PE + silicon) in the future, especially when high-performance computing is required [10].

In this thesis, several computing paradigms for EGT-based inkjet-printed technology were evaluated. It is believed that the research findings will help circuit designers to develop computing systems for future applications in the PE domain. The pertinent parameters such as circuit delay, power consumption or area usage are provided and enable the extrapolation to large-scale printed designs. Whenever possible, the proposed printed hardware was also validated on the system level by deployment of architecture-level simulations. For the machine learning classifiers, popular benchmark datasets were utilized for validating the circuit functionality and design concepts. To the best of my knowledge, for the first time an inkjet-printed latch, lookup-table, analog artificial neuron, decision trees and ROM element was fabricated in inkjet-printing technology. It is important to note, that all hardware prototypes in this work were optimized with respect to functionality, to obtain a proof of concept. However, substantial improvements are expected when optimizing for area usage, power consumption and delay.

### 6.3. Outlook

Besides the many contributions of this thesis, which already indicated the potential of inkjet-printing technologies for future application domains and the consumer goods market, several additional tasks have to be carried out to allow the application of EGT-based printed hardware on the industrial level.

As the passive conductive structures in this thesis were obtained by sputtering and laser ablation (Section 2.4), they have to be substituted by printed conductive materials to obtain an all-inkjet-printed process. Also the annealing step for the semiconductor ink has to be substituted in the EGT fabrication process, to allow for a fully room-temperature process. In this regard, research effort is made by using for instance UV-curing. By having both an all-inkjet- and room-temperature printing process, printed circuits can be fully manufactured by small desktop printers similar to the devices presented in this thesis. It is important to note, that the resolution of the ITO-sputtered conductive tracks is within the resolution of the inkjet-printing process deployed in this work. Thus, when moving to an all-inkjet-printing process, no increase in area usage of the presented designs is expected.

Further research will be centered around the reduction of process variations of EGTs and passive components to increase the chip yield of printed circuits. This enables large-scale designs with much more printed EGTs, beyond the hardware prototypes of this work. This will then allow the scalability of EGT-based designs and the possibility to have a fully functional demonstrator with integrated sensors and actuators.

Moreover, as analyzed in this thesis, analog circuit designs can substantially improve area usage and power consumption of printed circuits compared to conventional digital designs. However, automation of an analog design flow is in general difficult or even not feasible and requires expert knowledge. In this regard, an analog design library can help designers to realize efficient printed hardware in the future.

## 6. *Summary, Conclusion and Outlook*

As the circuits of the proposed hardware prototypes in this thesis were fabricated on a glass substrate, exploration of optional carrier materials is of great interest as it allows for flexible printed hardware. This is an ongoing task and expected to be solved in the near future.

As additive printing technology enables a layer-by-layer deposition technique, 3D-stacking of printed electronics is in general supported. Especially inkjet printing, which is a contact-less printing process, allows for stacking of functional materials. This can be leveraged in the future to increase the comparable low functional densities of printed hardware.

It is expected that EGT-technology will enable the penetration of before inaccessible fields of application by providing consumer electronics, which have unconventional conformity and cost requirements, which are beyond the capabilities of silicon-based technologies. Especially the implementation of printed machine learning classifiers for near-sensor-processing tasks is anticipated to play a key role in future application domains. It is believed that the contributions of this thesis give circuit designer for PE helpful insights for designing future printed electronic systems.

# Appendices





# A. Fabrication of EGTs

## A.1. Printing Steps

For the printed circuits a 20mm×20mm ITO-sputtered glass substrate (PGO CEC020S) which has a sheet resistance of  $20\Omega/\square$  and a layer thickness of 100nm was structured by laser ablation with a Trumpf TruMicro5000 picosecond laser. The infra-red laser wavelength was 1030nm with 2.5W laser power. After laser ablation of the ITO glass substrate, the passive conductive tracks were obtained. As an alternative, for one design (SR-latch) the passive conductive tracks were produced by structuring of the ITO substrate using eBeam lithography with poly(methyl methacrylate) PMMA as a photoresist.

Afterwards, the substrate was cleaned in an ultrasonic bath (Sonorex Digital 10P) for 20 minutes with a solution containing 50% Aceton and 50% Isopropanol. In a subsequent step, surface treatment was performed with an oxygen plasma cleaner (Diener Electronic) process for 2min.

The EGT fabrication starts with first inkjet-printing the semiconductor ink between the source and drain electrodes of the ITO conductive tracks. The substrate was then annealed at 400°C for 2 hours in a muffle furnace (Nabertherm P330), with a two hours ramp from room temperature to 400°C. After the annealing, the sheet resistance of the ITO conductive tracks increased to about  $80\Omega/\square$ . In addition, the semiconductor  $\text{In}_2\text{O}_3$  was obtained after annealing. After the cool-down phase, the CSPE was printed over the source, drain and gate ITO electrodes. Next, PEDOT:PSS was printed between the electrolyte and ITO gate electrode to obtain a top-gate contact. These EGT printing steps are also illustrated in Figure 2.7 and the EGT material stack is shown in Figure 2.6.

## A.2. Ink Preparation

For the semiconductor ink,  $\text{In}_2(\text{NO}_3)_3$  (Sigma-Aldrich, 99.9% trace metal basis, Indium (III) nitrate hydrate,  $\text{MW} = 300.83 \text{ g mol}^{-1}$ ) was dissolved in glycerol (Merck KGaA,  $\text{MW} = 92.09 \text{ g mol}^{-1}$ ) and double-deionized water with a ratio of 1:4 [81]. After stirring for 1 hour, the solution was filtered with 0.2 $\mu\text{m}$  polyvinylidene fluoride (PVDF) syringe filter. Next the ink was injected in the printer cartridge ink reservoir using a syringe.

For the electrolyte (CSPE) ink, 0.3g PVA (98% poly(vinyl alcohol) hydrolysed) was dissolved in 6g DMSO (Sigma-Aldrich, 99.9%, Dimethyl sulfoxide anhydrous,  $\text{MW} = 78.13 \text{ g mol}^{-1}$ ), and stirred for 2 hours at 90°C [81]. In addition, 0.07g of  $\text{LiClO}_4$ , lithium perchlorate (Sigma Aldrich, 99.99% trace metal basis,  $\text{MW} = 106.39 \text{ g mol}^{-1}$ ) was dissolved in 0.63g PC (Sigma-Aldrich, 99.7%, Propylene carbonate anhydrous,  $102.09 \text{ g mol}^{-1}$ ) and stirred for 1 hour at room temperature. Subsequently, both solutions were mixed and stirred until a clear solution was obtained. Finally, the solution was filtered with a 0.2 $\mu\text{m}$  PTFE (polytetrafluoroethylene)

### A. Fabrication of EGTs

syringe filter before injecting it into the cartridge ink reservoir.

For the PEDOT:PSS conductive ink, 70% PEDOT:PSS (Sigma-Aldrich, Poly(3,4-ethylenedioxythiophene) polystyrene sulfonate, 3.0 - 4.0% in H<sub>2</sub>O) were mixed with 30% ethylene glycol (Sigma-Aldrich, ethylene glycol anhydrous, 99%, MW = 62.07 g mol<sup>-1</sup>) [81]. The solution was stirred at room temperature until a clear solution was obtained. The solution was then filtered by 0.2µm polyvinylidene fluoride (PVDF) syringe filter before injecting it in the printer cartridge ink reservoir.

## B. Resistance to Neural Network Weight Calculation

The output voltage  $V_x$  of the crossbar architecture in Figure 4.3 depends on the input voltages applied to the resistors ( $V_i$  or  $\bar{V}_i$  (inverted inputs)) and the crossbar resistors  $R_i$ ,  $R_b$  and  $R_d$ . These crossbar resistors form a Y-circuit with only linear devices (resistors), thus the output signal  $V_x$  can be solved analytically.

The currents through all resistors are computed by Ohm's law:  $I_i = (V_i - V_x)/R_i = (V_i - V_x) g_i$ , and  $I_b = (V_{bias} - V_x)/R_b = (V_{bias} - V_x) g_b$  for the bias resistor  $R_b$  and  $I_d = (0V - V_x)/R_d = -V_x g_d$ , respectively.

These currents are now summed up according to Kirchhoff's rule:

$$\sum_{i=1}^P I_i + I_b + I_d = 0. \quad (\text{B.1})$$

By substituting the calculated currents into Equation (B.1), and by solving for  $V_x$ , the following equation is obtained:

$$V_x = \frac{\left(\sum_i \frac{V_i}{R_i}\right) + \frac{V_{bias}}{R_b}}{\left(\sum_i \frac{1}{R_i}\right) + \frac{1}{R_b} + \frac{1}{R_d}}. \quad (\text{B.2})$$

To obtain the relation between the hardware and model level of the NN weights, Equation (B.2) can be rewritten in the following form:

$$\begin{aligned} V_x &= \frac{\left(\sum_i \frac{V_i}{R_i}\right) + \frac{V_{bias}}{R_b}}{\left(\sum_i \frac{1}{R_i}\right) + \frac{1}{R_b} + \frac{1}{R_d}} \\ &= \sum_i V_i w_i + V_b w_b \\ &= \sum_i V_i w_i + b. \end{aligned} \quad (\text{B.3})$$

With the synaptic weights abbreviated by:

## B. Resistance to Neural Network Weight Calculation

$$w_i = \frac{\frac{1}{R_i}}{\left(\sum_j \frac{1}{R_j}\right) + \frac{1}{R_b} + \frac{1}{R_d}} \quad (\text{B.4})$$

$$= \frac{g_i}{\left(\sum_j g_j\right) + g_b + g_d}, \quad (\text{B.5})$$

and the bias weight:

$$w_b = \frac{\frac{1}{R_b}}{\left(\sum_j \frac{1}{R_j}\right) + \frac{1}{R_b} + \frac{1}{R_d}} \quad (\text{B.6})$$

$$= \frac{g_b}{\left(\sum_j g_j\right) + g_b + g_d}. \quad (\text{B.7})$$

Thus, the crossbar output  $V_x$  implements the MAC operation of artificial neural networks [58]:

$$a = V_x = \sum_i w_i V_i + b = \sum_i w_i x_i + b.$$

As can be obtained from Equation (B.4) and Equation (B.6), the NN weights are determined by the resistances of the printed crossbar resistors  $R_i$ . To achieve one-time programmability of the crossbar, different resistor geometries are printed according to the trained weights from the NN learning procedure.

# Glossary

- ADC** Analog-to-digital converter. 27, 38, 40, 45, 67, 70, 76
- AND** AND gate. xiii, 20, 22, 25
- ANN** Artificial neural network. xi, xiii–xv, 4, 5, 28–30, 32, 33, 35, 36, 38, 40, 42–45, 47–49, 51, 53–57, 74, 75
- CAD** Computer-aided design. 7, 70
- CMOS** Complementary metal-oxide-semiconductor. 4, 18, 19, 77
- CPU** Central processing unit. 28, 29
- CSPE** Composite solid polymer electrolyte. 12–14
- DC** Direct current. 15, 35
- EGT** Electrolyte-gated Field Effect Transistor. vii–x, xiii, 1–7, 12–16, 18, 19, 22, 25, 27, 32, 35, 36, 48, 50, 54, 63, 64, 69, 70, 73–78
- FPGA** Field-programmable gate array. 4, 19
- FSM** Finite-state machine. 47, 48
- GPU** Graphics processing unit. 28
- IC** Integrated Circuit. 7
- inv** Inverter-based negative weights operation. xiii, 35, 38, 41, 42
- IoT** Internet of Things. vii, 1, 44
- IT** Information Technology. vii, 1
- ITO** Indium tin oxide. 12, 13, 16, 22, 30, 36
- LSB** Least significant bit. 62
- LUT** Lookup Table. xiii, xiv, 4, 6, 19, 20, 22, 24, 26, 27, 44, 64, 73, 74
- MAC** Multiply-accumulate operation or multiplier-accumulator. xiii, 4, 28–33, 35, 36, 38, 40–42, 44, 45, 47, 48, 51, 52, 55, 74
- MSB** Most significant bit. 62
- NAND** "Not AND" gate. 13, 20, 61

- NCS** Neuromorphic computing system. 27, 28, 38, 40, 43, 44, 48
- NN** Neural Network. 43, 44
- NOR** "Not OR" gate. 13, 15, 20, 61
- OR** OR gate. 20
- PE** Printed Electronics. vii, xiii, 1–4, 6, 7, 13, 19, 27–29, 38, 43–45, 48, 49, 56, 57, 60, 68, 69, 75–78
- PEDOT:PSS** Poly(3,4-ethylenedioxythiophene) polystyrene sulfonate. 13, 30, 33, 64
- pLUT1** Printed 1-input Lookup Table. 19–22
- pLUT2** Printed 2-input Lookup Table. 20–22
- PMMA** Poly(methyl methacrylate). 13
- pNeuron** Printed artificial neuron. 38, 41
- PPDK** Printed process design kit. 15, 35, 36
- pPLU** Printed piece-wise linear unit. 32, 33, 36
- ptanh** Printed hyperbolic tangent. xiii, 36, 38, 39, 41, 43
- ReLU** Rectified linear unit. 38, 40, 47, 50–54, 57
- ROM** Read-only memory. xiv, 5, 6, 59, 69–71, 73, 74, 77
- SC** Stochastic Computing. xiv, xv, 5, 45–50, 53–57, 74, 75
- SC-NN** Stochastic computing-based neural networks. xiv, xv, 45, 47, 49, 51, 54–56, 74, 75
- SNG** Stochastic number generator. 45, 48, 49, 52–54
- SNN** Spiking neural network. 28
- tanh** Hyperbolic tangent. 4
- TRNG** True random number generator. 48
- VLSI** Very large scale integration. 7
- XNOR** "Exclusive NOR" gate. xiii, 5, 20–23, 45, 46, 48, 51, 54
- XOR** "Exclusive OR" gate. xiii, 13, 20, 22, 24

# Bibliography

- [1] S. Matsuoka, H. Amano, K. Nakajima, K. Inoue, T. Kudoh, N. Maruyama, K. Taura, T. Iwashita, T. Katagiri, T. Hanawa *et al.*, “From flops to bytes: disruptive change in high-performance computing towards the post-moore era,” in *Proceedings of the ACM International Conference on Computing Frontiers*, pp. 274–281, 2016.
- [2] M. M. Waldrop, “More than moore,” *Nature*, vol. 530, no. 7589, pp. 144–148, 2016.
- [3] J. A. Spaeder, “Radio frequency identification pharmaceutical tracking system and method,” 2005.
- [4] M. Peris and L. Escuder-Gilabert, “Electronic noses and tongues to assess food authenticity and adulteration,” *Trends in Food Science & Technology*, vol. 58, pp. 40–54, 2016.
- [5] J. A. Cavallo, M. C. Strumia, and C. G. Gomez, “Preparation of a milk spoilage indicator adsorbed to a modified polypropylene film as an attempt to build a smart packaging,” *Journal of Food Engineering*, vol. 136, pp. 48 – 55, 2014.
- [6] Y. J. Kim, S.-E. Chun, J. Whitacre, and C. J. Bettinger, “Self-deployable current sources fabricated from edible materials,” *Journal of Materials Chemistry B*, vol. 1, no. 31, pp. 3781–3788, 2013.
- [7] A. Chortos, J. Liu, and Z. Bao, “Pursuing prosthetic electronic skin,” *Nature materials*, vol. 15, no. 9, pp. 937–950, 2016.
- [8] Y. Lee, J. Y. Oh, W. Xu, O. Kim, T. R. Kim, J. Kang, Y. Kim, D. Son, J. B.-H. Tok, M. J. Park *et al.*, “Stretchable organic optoelectronic sensorimotor synapse,” *Science advances*, vol. 4, no. 11, p. eaat7387, 2018.
- [9] Y. Kim, A. Chortos, W. Xu, Y. Liu, J. Y. Oh, D. Son, J. Kang, A. M. Foudeh, C. Zhu, Y. Lee *et al.*, “A bioinspired flexible organic artificial afferent nerve,” *Science*, vol. 360, no. 6392, pp. 998–1003, 2018.
- [10] Y. Khan, A. Thielens, S. Muin, J. Ting, C. Baumbauer, and A. C. Arias, “A new frontier of printed electronics: flexible hybrid electronics,” *Advanced Materials*, vol. 32, no. 15, p. 1905279, 2020.
- [11] R. a. Markets, “Global fmcg market opportunity analysis, 2018-2019 2025 - a billion new consumers in emerging markets,” Sep 2019. [Online]. Available: <https://www.prnewswire.com/news-releases/global-fmcg-market-opportunity-analysis-2018-2019--2025---a-billion-new-consumers-in-emerging-markets-300913934.html>
- [12] J. Handy, “Why are computer chips so expensive?” <https://www.forbes.com/sites/jimhandy/2014/04/30/why-are-chips-so-expensive/#3ee0f8f479c9>, 2014.
- [13] L. Liu, Y. Feng, and W. Wu, “Recent progress in printed flexible solid-state supercapacitors for portable and wearable energy storage,” *Journal of Power Sources*, vol. 410, pp. 69–77, 2019.

## BIBLIOGRAPHY

- [14] J. S. Chang, A. F. Facchetti, and R. Reuss, “A circuits and systems perspective of organic/printed electronics: Review, challenges, and contemporary and emerging design approaches,” *IEEE Journal on emerging and selected topics in circuits and systems*, vol. 7, no. 1, pp. 7–26, 2017.
- [15] B. Shao, “Fully printed chipless rfid tags towards item-level tracking applications,” Ph.D. dissertation, KTH Royal Institute of Technology, 2014.
- [16] Y. Huang, H. Wu, L. Xiao, Y. Duan, H. Zhu, J. Bian, D. Ye, and Z. Yin, “Assembly and applications of 3d conformal electronics on curvilinear surfaces,” *Materials Horizons*, vol. 6, no. 4, pp. 642–683, 2019.
- [17] S. Mühl and B. Beyer, “Bio-organic electronics—overview and prospects for the future,” *Electronics*, vol. 3, no. 3, pp. 444–461, 2014.
- [18] J. Noh, M. Jung, Y. Jung, C. Yeom, M. Pyo, and G. Cho, “Key issues with printed flexible thin film transistors and their application in disposable rf sensors,” *Proceedings of the IEEE*, vol. 103, no. 4, pp. 554–566, 2015.
- [19] A. J. Bhandodkar, W. Jia, and J. Wang, “Tattoo-based wearable electrochemical devices: a review,” *Electroanalysis*, vol. 27, no. 3, pp. 562–572, 2015.
- [20] M. Douthwaite, F. García-Redondo, P. Georgiou, and S. Das, “A time-domain current-mode mac engine for analogue neural networks in flexible electronics,” in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4, 2019.
- [21] Y. Takeda, K. Hayasaka, R. Shiwaku, K. Yokosawa, T. Shiba, M. Mamada, D. Kumaki, K. Fukuda, and S. Tokito, “Fabrication of ultra-thin printed organic tft cmos logic circuits optimized for low-voltage wearable sensor applications,” *Scientific reports*, vol. 6, no. 1, pp. 1–9, 2016.
- [22] Y. T. et al., “Organic cmos logic circuit,” [https://commons.wikimedia.org/wiki/File:Organic\\_CMOS\\_logic\\_circuit.jpg](https://commons.wikimedia.org/wiki/File:Organic_CMOS_logic_circuit.jpg), 2016, licence: CC BY (<https://creativecommons.org/licenses/by/4.0>).
- [23] Z. Cui, *Printed electronics: materials, technologies and applications*. John Wiley & Sons, 2016.
- [24] S. K. Garlapati, M. Divya, B. Breitung, R. Kruk, H. Hahn, and S. Dasgupta, “Printed electronics based on inorganic semiconductors: from processes and materials to devices,” *Advanced Materials*, vol. 30, no. 40, p. 1707600, 2018.
- [25] S. K. Garlapati, N. Mishra, S. Dehm, R. Hahn, R. Kruk, H. Hahn, and S. Dasgupta, “Electrolyte-gated, high mobility inorganic oxide transistors from printed metal halides,” *ACS applied materials & interfaces*, vol. 5, no. 22, pp. 11 498–11 502, 2013.
- [26] B. J. MacLennan, “The promise of analog computation,” *International Journal of General Systems*, vol. 43, no. 7, pp. 682–696, 2014.
- [27] F. Rasheed, M. Hefenbrock, M. Beigl, M. B. Tahoori, and J. Aghassi-Hagmann, “Variability modeling for printed inorganic electrolyte-gated transistors and circuits,” *IEEE Transactions on Electron Devices*, vol. 66, no. 1, pp. 146–152, 2018.
- [28] D. Weller, G. C. Marques, J. Aghassi-Hagmann, and M. B. Tahoori, “An inkjet-printed low-voltage latch based on inorganic electrolyte-gated transistors,” *IEEE Electron Device*



- Letters*, vol. 39, no. 6, pp. 831–834, 2018.
- [29] A. T. Erozan, D. D. Weller, F. Rasheed, R. Bishnoi, J. Aghassi-Hagmann, and M. B. Tahoori, “A novel printed look-up table based programmable printed digital circuit,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.
- [30] D. D. Weller, M. Hefenbrock, M. B. Tahoori, J. Aghassi-Hagmann, and M. Beigl, “Programmable neuromorphic circuit based on printed electrolyte-gated transistors,” in *Proceedings of the Asia South Pacific design automation conference (ASP-DAC)*, 2020.
- [31] D. D. Weller, M. Hefenbrock, M. Beigl, J. Aghassi-Hagmann, and M. B. Tahoori, “Realization and training of an inverter-based printed neuromorphic computing system,” in *Nature Scientific Report (submitted on March 2nd 2021)*, 2021.
- [32] D. D. Weller, N. Bleier, M. Hefenbrock, J. Aghassi-Hagmann, M. Beigl, R. Kumar, and M. B. Tahoori, “Printed stochastic computing neural networks,” in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE) (accepted)*, 2021.
- [33] M. H. Mubarik, D. D. Weller, N. Bleier, M. Tomei, J. Aghassi-Hagmann, M. B. Tahoori, and R. Kumar, “Printed machine learning classifiers,” in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 73–87, 2020.
- [34] V. S. et al., “Printed electronics for low-cost electronic systems: Technology status and application development,” in *2008-34th ESSCIRC*, pp. 17–24, 2008.
- [35] M. G. M. et al., “All-printed flexible and stretchable electronics,” *Advanced Materials*, vol. 29, no. 19, p. 1604965, 2017.
- [36] P.-Y. Chen, C.-L. Chen, C.-C. Chen, L. Tsai, H.-C. Ting, L.-F. Lin, C.-C. Chen, C.-Y. Chen, L.-H. Chang, T.-H. Shih *et al.*, “30.1: Invited paper: 65-inch inkjet printed organic light-emitting display panel with high degree of pixel uniformity,” in *SID Symposium Digest of Technical Papers*, vol. 45, no. 1, pp. 396–398, 2014.
- [37] J. Zhao, Y. Gao, W. Gu, C. Wang, J. Lin, Z. Chen, and Z. Cui, “Fabrication and electrical properties of all-printed carbon nanotube thin film transistors on flexible substrates,” *Journal of Materials Chemistry*, vol. 22, no. 38, pp. 20 747–20 753, 2012.
- [38] J. Chang, X. Zhang, T. Ge, and J. Zhou, “Fully printed electronics on flexible substrates: High gain amplifiers and dac,” *Organic Electronics*, vol. 15, no. 3, pp. 701–710, 2014.
- [39] W. Voit, W. Zapka, P. Dyreklev, O.-J. Hagel, A. Hägerström, and P. Sandström, “Inkjet printing of non-volatile rewritable memory arrays,” in *NIP & Digital Fabrication Conference*, vol. 2006, no. 3, pp. 34–37, 2006.
- [40] A. de la Fuente Vornbrock, D. Sung, H. Kang, R. Kitsomboonloha, and V. Subramanian, “Fully gravure and ink-jet printed high speed pbttt organic thin film transistors,” *Organic Electronics*, vol. 11, no. 12, pp. 2037–2044, 2010.
- [41] A. Almusallam, R. Torah, D. Zhu, M. Tudor, and S. Beeby, “Screen-printed piezoelectric shoe-insole energy harvester using an improved flexible pzt-polymer composites,” in *Journal of Physics: Conference Series*, vol. 476, no. 1, p. 012108, 2013.
- [42] W. J. Hyun, E. B. Secor, M. C. Hersam, C. D. Frisbie, and L. F. Francis, “High-resolution patterning of graphene by screen printing with a silicon stencil for highly flexible printed electronics,” *Advanced Materials*, vol. 27, no. 1, pp. 109–115, 2015.

## BIBLIOGRAPHY

- [43] M. Jung, J. Kim, J. Noh, N. Lim, C. Lim, G. Lee, J. Kim, H. Kang, K. Jung, A. D. Leonard *et al.*, “All-printed and roll-to-roll-printable 13.56-mhz-operated 1-bit rf tag on plastic foils,” *IEEE Transactions on Electron Devices*, vol. 57, no. 3, pp. 571–580, 2010.
- [44] T. Carey, S. Cacovich, G. Divitini, J. Ren, A. Mansouri, J. M. Kim, C. Wang, C. Ducati, R. Sordan, and F. Torrisi, “Fully inkjet-printed two-dimensional material field-effect heterojunctions for wearable and textile electronics,” *Nature communications*, vol. 8, no. 1, pp. 1–11, 2017.
- [45] S. Conti, L. Pimpolari, G. Calabrese, R. Worsley, S. Majee, D. K. Polyushkin, M. Paur, S. Pace, D. H. Keum, F. Fabbri *et al.*, “Low-voltage 2d materials-based printed field-effect transistors for integrated digital and analog electronics on paper,” *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.
- [46] F. A. Viola, B. Brigante, P. Colpani, G. Dell’Erba, V. Mattoli, D. Natali, and M. Caironi, “A 13.56 mhz rectifier based on fully inkjet printed organic diodes,” *Advanced Materials*, p. 2002329, 2020.
- [47] W. Xiong, Y. Guo, U. Zschieschang, H. Klauk, and B. Murmann, “A 3-v, 6-bit c-2c digital-to-analog converter using complementary organic thin-film transistors on glass,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 7, pp. 1380–1388, 2010.
- [48] B. Huber, P. Popp, M. Kaiser, A. Ruediger, and C. Schindler, “Fully inkjet printed flexible resistive memory,” *Applied Physics Letters*, vol. 110, no. 14, p. 143503, 2017.
- [49] M. Kondo, T. Uemura, M. Akiyama, N. Namba, M. Sugiyama, Y. Noda, T. Araki, S. Yoshimoto, and T. Sekitani, “Design of ultraflexible organic differential amplifier circuits for wearable sensor technologies,” in *2018 IEEE International Conference on Microelectronic Test Structures (ICMTS)*, pp. 79–84, 2018.
- [50] G. Cadilha Marques, D. Weller, A. T. Erozan, X. Feng, M. Tahoori, and J. Aghassi-Hagmann, “Progress report on ”from printed electrolyte-gated metal-oxide devices to circuits”,” *Advanced Materials*, p. 1806483, 2019.
- [51] G. Cadilha Marques, S. K. Garlapati, S. Dehm, S. Dasgupta, H. Hahn, M. Tahoori, and J. Aghassi-Hagmann, “Digital power and performance analysis of inkjet printed ring oscillators based on electrolyte-gated oxide electronics,” *Applied Physics Letters*, vol. 111, no. 10, p. 102103, 2017.
- [52] S. Kiamehr, A. Amouri, and M. B. Tahoori, “Investigation of nbtI and pbti induced aging in different lut implementations,” in *2011 International Conference on Field-Programmable Technology*, pp. 1–8, 2011.
- [53] J. Narasimham, K. Nakajima, C. S. Rim, and A. T. Dahbura, “Yield enhancement of programmable asic arrays by reconfiguration of circuit placements,” *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 13, no. 8, pp. 976–986, 1994.
- [54] B. Liu, H. Li, Y. Chen, X. Li, Q. Wu, and T. Huang, “Vortex: variation-aware training for memristor x-bar,” in *Proceedings of the 52nd Annual Design Automation Conference*, p. 15, 2015.
- [55] C.-S. Leung, W. Y. Wan, and R. Feng, “A regularizer approach for rbf networks under the concurrent weight failure situation,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 6, pp. 1360–1372, 2016.

- [56] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. Pernice, “All-optical spiking neurosynaptic networks with self-learning capabilities,” *Nature*, vol. 569, no. 7755, pp. 208–214, 2019.
- [57] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, “A survey of neuromorphic computing and neural networks in hardware,” *arXiv preprint arXiv:1705.06963*, 2017.
- [58] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [59] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [60] Y. van De Burgt, A. Melianas, S. T. Keene, G. Malliaras, and A. Salleo, “Organic electronics for neuromorphic computing,” *Nature Electronics*, p. 1, 2018.
- [61] R. A. Nawrocki, R. M. Voyles, and S. E. Shaheen, “Neurons in polymer: Hardware neural units based on polymer memristive devices and polymer transistors,” *IEEE Transactions on Electron Devices*, vol. 61, no. 10, pp. 3513–3519, 2014.
- [62] M. Ansari, A. Fayyazi, A. Banagozar, M. A. Maleki, M. Kamal, A. Afzali-Kusha, and M. Pedram, “Phax: Physical characteristics aware ex-situ training framework for inverter-based memristive neuromorphic circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1602–1613, 2017.
- [63] N. Bleier, M. Mubarak, F. Rasheed, J. Aghassi-Hagmann, M. Tahoori, and R. Kumar, “Printed microprocessors,” in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020.
- [64] K. Crammer and Y. Singer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *Journal of machine learning research*, vol. 2, no. Dec, pp. 265–292, 2001.
- [65] D. D. et al., “UCI machine learning repository,” 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [66] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS-W*, 2017.
- [67] D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, “Physics for neuromorphic computing,” *Nature Reviews Physics*, pp. 1–12, 2020.
- [68] S. Gong, W. Schwalb, Y. Wang, Y. Chen, Y. Tang, J. Si, B. Shirinzadeh, and W. Cheng, “A wearable and highly sensitive pressure sensor with ultrathin gold nanowires,” *Nature communications*, vol. 5, no. 1, pp. 1–8, 2014.
- [69] P. He, J. Brent, H. Ding, J. Yang, D. Lewis, P. O’Brien, and B. Derby, “Fully printed high performance humidity sensors based on two-dimensional materials,” *Nanoscale*, vol. 10, no. 12, pp. 5599–5606, 2018.
- [70] L. Nazhandali, B. Zhai, A. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, T. Austin, and D. Blaauw, “Energy optimization of subthreshold-voltage sensor network processors,” in *32nd International Symposium on Computer Architecture (ISCA ’05)*, pp. 197–207, 2005.
- [71] J. Kim, I. Jeerapan, S. Imani, T. N. Cho, A. Bandodkar, S. Cinti, P. P. Mercier, and J. Wang, “Noninvasive alcohol monitoring using a wearable tattoo-based iontophoretic-biosensing system,” *Acs Sensors*, vol. 1, no. 8, pp. 1011–1019, 2016.

## BIBLIOGRAPHY

- [72] P. Mostafalu, W. Lenk, M. R. Dokmeci, B. Ziaie, A. Khademhosseini, and S. R. Sonkusale, “Wireless flexible smart bandage for continuous monitoring of wound oxygenation,” *IEEE Transactions on biomedical circuits and systems*, vol. 9, no. 5, pp. 670–677, 2015.
- [73] A. Alaghi, C. Li, and J. P. Hayes, “Stochastic circuits for real-time image-processing applications,” in *Proceedings of the 50th Annual Design Automation Conference*, pp. 1–6, 2013.
- [74] Y. Liu and K. K. Parhi, “Architectures for recursive digital filters using stochastic computing,” *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3705–3718, 2016.
- [75] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, “An architecture for fault-tolerant computation with stochastic logic,” *IEEE transactions on computers*, vol. 60, no. 1, pp. 93–105, 2010.
- [76] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, “Vlsi implementation of deep neural network using integral stochastic computing,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2688–2699, 2017.
- [77] A. Ren, Z. Li, C. Ding, Q. Qiu, Y. Wang, J. Li, X. Qian, and B. Yuan, “Sc-dcnn: Highly-scalable deep convolutional neural network using stochastic computing,” *ACM SIGPLAN Notices*, vol. 52, no. 4, pp. 405–418, 2017.
- [78] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, “A survey of stochastic computing neural networks for machine learning applications,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [79] B. R. Gaines, “Stochastic computing,” in *Proceedings of the April 18-20, 1967, spring joint computer conference*, pp. 149–156, 1967.
- [80] V. Canals, A. Morro, A. Oliver, M. L. Alomar, and J. L. Rosselló, “A new stochastic computing methodology for efficient neural network implementation,” *IEEE transactions on neural networks and learning systems*, vol. 27, no. 3, pp. 551–564, 2015.
- [81] G. C. Marques, F. von Seggern, S. Dehm, B. Breitung, H. Hahn, S. Dasgupta, M. B. Tahoori, and J. Aghassi-Hagmann, “Influence of humidity on the performance of composite polymer electrolyte-gated field-effect transistors and circuits,” *IEEE Transactions on Electron Devices*, vol. 66, no. 5, pp. 2202–2207, 2019.
- [82] A. T. Erozan, G. Y. Wang, R. Bishnoi, J. Aghassi-Hagmann, and M. B. Tahoori, “A compact low-voltage true random number generator based on inkjet printing technology,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 6, pp. 1485–1495, 2020.
- [83] L. B. et al., *Classification and regression trees*, ser. The Wadsworth statistics/probability series. Monterey, CA: Wadsworth Brooks/Cole Advanced Books Software, 1984.
- [84] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.
- [85] T. H. et al., *The elements of statistical learning: prediction, inference and data mining*, 2nd ed. Springer, 2009.
- [86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,”

*Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

- [87] E. De Angel and E. E. Swartzlander Jr, “Survey of low power techniques for roms,” in *Proceedings of the 1997 international symposium on Low power electronics and design*, pp. 7–11, 1997.
- [88] D. A. Rich, “A survey of multivalued memories,” *IEEE Transactions on Computers*, no. 2, pp. 99–106, 1986.

