

[Download full-text PDF](#) [Download citation](#) [Copy link](#)[Thesis](#) [PDF Available](#)**Beyond 0 and 1: A mixed radix design and verification workflow for modern ternary computers**

May 2024

DOI:[10.13140/RG.2.2.21138.57289](https://doi.org/10.13140/RG.2.2.21138.57289)

Thesis for: PhD · Advisor: Henning Gundersen and Nils-Olav Skeie

Authors:**Steven Bos**

University of South-Eastern Norway

[Citations \(1\)](#)[References \(306\)](#)**Abstract**

For more than 80 years digital computers use the radix-2 or binary computer alphabet as their lowest symbolic and physical representation. This doctrine of computing is presumed in every modern computer. The radix economy theorem derives that radix-3 or ternary is however the optimal radix. Ternary is the first radix in the Multiple-Valued Logic (MVL) family that enables symmetrical arithmetic using the balanced ternary notation. The ongoing challenge is to engineer devices, circuits and systems that can physically represent three logic levels with competitive power, performance, area and cost metrics. For flash storage and communication MVL is already the industry standard, but logic remains binary. Ever since Dennard scaling stopped in 2005, binary computing is struggling to overcome the increasing power wall, memory wall and Electronic Design and Automation (EDA) wall. A unified MVL compute paradigm can theoretically address these challenges, making it a prime candidate for the beyond-CMOS era. This article-based thesis is structured in three parts. In the first part binary computing is discussed. The historical reasoning for this choice as well as the current scaling challenges that impede its future were reviewed. The part concludes with a review of several fundamental and engineering limits that are rarely cited but highly relevant when considering another radix such as Shannon's noisy channel theorem and Rent's rule. In the second part ternary computing is discussed. A brief overview of radix-3 theory and literature is presented. A novel radix comparison methodology is proposed to improve fairness. Historical efforts to build ternary computers were reviewed which started in the 1950's. A categorization of the main benefits of balanced ternary is presented across 7 application domains. The part concludes with an overview of the critique on radix-3. In the third part practical aspects of ternary computing are discussed: multi-stable devices and EDA tooling. For devices, non-volatile ternary memory control with commercially available memristors was studied. A novel open source software tool uMemristorToolbox and hardware platform for multi-state memristor programming were developed. The experiments confirm that ternary memory with memristors is both feasible and low-cost. Lastly, EDA tooling and workflows for ternary logic chips are discussed. The open source software tool Mixed Radix Circuit Synthesizer (MRCS) was developed, the first browserbased EDA tool to design and verify binary, ternary and hybrid (mixed radix) circuits. It features a novel MVL circuit synthesis algorithm with HSPICE and verilog output targeting CMOS and multi-threshold CNTFET. The tool was used to design REBEL-2, a novel balanced ternary CPU with RISC-V-like ISA. Four MRCS designs have been tested on a FPGA and submitted for tape-out using the Openlane ASIC workflow.

Discover the world's research

- 25+ million members
- 160+ million publication pages
- 2.3+ billion citations

[Join for free](#) [Public Full-text \(1\)](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Steven Bos

Beyond 0 and 1:

A mixed radix design and verification workflow
for modern ternary computers

Dissertation for the
degree of Ph.D
Technology

Faculty of Technology, Natural
Sciences and Maritime Studies

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Steven Bos

Beyond 0 and 1:

A mixed radix design and verification workflow for modern ternary computers

A PhD dissertation in
Technology

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

© Steven Bos, 2024

Faculty of Technology, Natural Sciences and Maritime Studies
University of South-Eastern Norway
Kongsberg

Doctoral dissertations at the University of South-Eastern Norway no.189

ISSN: 2535-5244 (print)

ISSN: 2535-5252 (online)

ISBN: 978-82-7206-854-6 (print)

ISBN: 978-82-7206-855-3 (online)

This publication is licensed with a Creative Commons license. You may copy and redistribute the material in any medium or format. You must give appropriate credit, provide a link to the license, and indicate if changes were made. Complete license terms at <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

Print: University of South-Eastern Norway

To my heroes Koen & Susan Bos-Theuvenet

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

I

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

II

Preface

This dissertation is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) in Technology from the Department of Science and Industry Systems at the University of South-Eastern Norway (USN). The doctoral work presented here took place between June 1, 2019 and June 1, 2023. The majority of the research was conducted at USN campus Kongsberg in Norway and with some work done at USN campuses Vestfold and Porsgrunn. The work has been done under the supervision of Assoc. Professor Henning Gundersen (USN) and Professor Nils-Olav Skeie (USN). Further guidance was received from the midterm evaluation committee Professor Philipp D. Häfliger from the University of Oslo (UiO) and Professor Lars M. Johansen (USN).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Research (Kunnskapsdepartementet) and USN as a 4 year PhD Research Fellowship (KFD-stilling) with 25% teaching duties.

The candidate is a member of the IEEE MVL and CAS societies as well as the research school for Training the Next Generation of Micro- and Nanotechnology Researchers in Norway (TNNN).

Acknowledgements

During my four year PhD journey I had the pleasure of meeting wonderful and inspiring people and perhaps a new version of myself. Moving from The Netherlands to Norway with my wife and learning the Norwegian language and rich culture was not possible without a long list of people mentioned below – *mea culpa* if I missed some of you!

Without my supervisors Assoc. Professor Henning Gundersen and Professor Nils-Olav Skeie my adventure in the world of ternary computing would not have started. I am deeply grateful for the time they spend with me.

Research is not one solitary endeavour nor does it start at zero as we always build on the insights made before us. Working closely with my fellow co-authors was a true privilege and I like to thank them again for their contributions. Collaborating with Professor Knut E. Aasmundtveit and Assoc. Professor Avisek Roy was immensely valuable as I was able learn about the possibilities of carbon nanotubes and nanofabrication at USN. Thank you for sharing your knowledge and I look forward to continuing our collaboration.

The knowledge landscape is shaped by giants and discussing my research on multi-state memristors with the inventor of these devices, Professor Leon Chua from UC Berkeley was extraordinary. At the start of my PhD I stumbled upon the work of professor Kris

III

Campbell from Boise State University on programming memristors. Her detailed work inspired me to investigate the analog switching properties of resistive memory. I also like to thank the many great minds I met both in-person and virtually due to the COVID pandemic.

A huge thanks to Professor Morten Melaaen (dean), Assoc. Professor Elisabet Syverud (head of department), Professor Olaf Hallan Graven (former head of department) and Rune Romnes for providing me with a great work environment and resources. Thanks to the USN PhD committee and PhD coordinators Mariken Kjøhl-Røsand, Per Morten Hansen and Siri Luise Tveitan for guidance, structure and organizing the inspiring PhD forums. I must mention the colleagues I had the pleasure to teach three courses with the

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Starting my work in Norway, I would like to thank my advisor Prof. Rigmor Baraas for teaching me the nuances in bokmål and nynorsk and Karoline Moholt McClenaghan for sharing her thrilling stories about the first computers in Norway. I am very grateful to Professor Rigmor Baraas from the department of Optometry to allow me to present my PhD work to the Norwegian prime minister Jonas Gahr Støre.

Starting a new research group together with Assoc. Prof. Henning Gunderson brought structure to my work. Meeting weekly with members of the Ternary Research Group, discussing and disseminating the tiniest results was a true joy. I like to thank all members present and past for input on papers and their open-mindedness to explore an unconventional computing paradigm. I especially enjoyed the boundless energy of dr. Radmila Juric.

I had the opportunity to supervise four bright MSc students Halvor Nybø Risto (now PhD candidate), Julian Breivold Nilsen, Mehtab Singh Virk and Erika Fegri and assisted several BSc groups. Thanks for sharing the long hours in pursuit of the adrenaline rush of inevitable progress. I like to thank post-doc dr. Fahim Ahmed Salim and my fellow PhD candidates for an awesome time: Walter Kibet Yego, Haytham Ali, Rune Andre Haugen, Tommy Langen, Agnieszka Lach, Soheila Taghavi Hosnaroudi and Raghav Sikka.

Sharing my work with my old and new friends, brother Niels and sister Jenna-Fay and the always curious Bos and Theuvenet family was immensely relevant. It forced me to find new metaphors and take different viewpoints. Thank you for your support!

This dissertation would not be possible without the unwavering support and extreme patience of my wife Jessica Stokhof. I am hugely indebted and will start returning the immense favor with this ternary thank you: 3^{thank you}

Steven Bos

Kongsberg, 1st November 2023

IV

Abstract

For more than 80 years digital computers use the radix-2 or *binary* computer alphabet as their lowest symbolic and physical representation. This doctrine of computing is presumed in every modern computer. The radix economy theorem derives that radix-3 or *ternary* is however the optimal radix. Ternary is the first radix in the Multiple-Valued Logic (MVL) family that enables symmetrical arithmetic using the balanced ternary notation. The

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

storage and communication MVL is already the industry standard, but logic remains binary. Ever since Dennard scaling stopped in 2005, binary computing is struggling to overcome the increasing power wall, memory wall and Electronic Design and Automation (EDA) wall. A unified MVL compute paradigm can theoretically address these challenges, making it a prime candidate for the beyond-CMOS era.

This article-based thesis is structured in three parts. In the first part binary computing is discussed. The historical reasoning for this choice as well as the current scaling challenges that impede its future were reviewed. The part concludes with a review of several fundamental and engineering limits that are rarely cited but highly relevant when considering another radix such as Shannon's noisy channel theorem and Rent's rule.

In the second part ternary computing is discussed. A brief overview of radix-3 theory and literature is presented. A novel radix comparison methodology is proposed to improve fairness. Historical efforts to build ternary computers were reviewed which started in the 1950's. A categorization of the main benefits of balanced ternary is presented across 7 application domains. The part concludes with an overview of the critique on radix-3.

In the third part practical aspects of ternary computing are discussed: multi-stable devices and EDA tooling. For devices, non-volatile ternary memory control with commercially available memristors was studied. A novel open source software tool uMemristorToolbox and hardware platform for multi-state memristor programming were developed. The experiments confirm that ternary memory with memristors is both feasible and low-cost.

Lastly, EDA tooling and workflows for ternary logic chips are discussed. The open source software tool Mixed Radix Circuit Synthesizer (MRCS) was developed, the first browser-based EDA tool to design and verify binary, ternary and hybrid (mixed radix) circuits. It features a novel MVL circuit synthesis algorithm with HSPICE and verilog output targeting CMOS and multi-threshold CNTFET. The tool was used to design REBEL-2, a novel balanced ternary CPU with RISC-V-like ISA. Four MRCS designs have been tested on a FPGA and submitted for tape-out using the Openlane ASIC workflow.

Keywords: ternary microprocessor, design automation, integrated circuit synthesis

V

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

VI

Contents

Preface

III

[Download full-text PDF](#)[Download citation](#)[Copy link](#)**Contents**

	IX
List of Papers	XI
List of Co-supervised Projects	XIII
List of Figures	XVI
List of Tables	XVII
Nomenclature	XIX
1 Introduction	1
1.1 The computer alphabet	1
1.2 Motivation and scope	2
1.3 Historical evolution of binary computing	6
1.4 Moore's curse: 3 scaling walls	9
1.4.1 Power wall	10
1.4.2 Memory wall	11
1.4.3 EDA wall	14
1.5 The fundamental limits of computing	15
1.5.1 Shannon's limit	16
1.5.2 Landauer's limit	19
1.5.3 Radix economy	20
1.5.4 Rent's rule	23
1.6 Research objective and dissertation structure	24
2 The benefits of ternary	27
2.1 Introduction	27
2.2 Ternary basics	29
2.2.1 Heptavintimal notation	29
2.2.2 The third value	30
2.2.3 Mixed radix	31
2.3 Radix comparison methodology	32
2.4 Historical evolution of ternary computing	34
2.5 The seven C's of ternary	36
2.5.1 Computation	36
2.5.2 Communication	37
2.5.3 Energy Consumption	38
2.5.4 Compression	39

VII

Contents

2.5.5 Comprehension	40
2.5.6 Cyber-Security	40
2.5.7 Design complexity	40

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

3 Multi-state RRAM development platform	45
3.1 Introduction	45
3.2 Multi-state programming	46
3.3 uMemristorToolbox: A new tool for experimenting with multi-state RRAM	49
3.3.1 Motivation	49
3.3.2 Architecture	50
3.3.3 Experiments	51
3.3.4 Application: Embedded ternary system	54
3.4 Ternary memory controller circuit	55
3.4.1 Simulation	56
3.4.2 Implementation	56
3.5 Conclusion	57
4 Mixed radix EDA for ternary computers	59
4.1 Introduction	59
4.2 MRCS: A new tool for mixed radix design and verification	60
4.2.1 Motivation	60
4.2.2 Architecture	61
4.2.3 Workflows	63
4.3 Mixed radix synthesis engine	65
4.3.1 Introduction	65
4.3.2 Related work	66
4.3.3 Mixed radix synthesis algorithm	67
4.3.4 Binary coded ternary RTL	69
4.4 REBEL-2 Balanced Ternary CPU	71
4.4.1 Motivation	72
4.4.2 Balanced Ternary Instruction Set Architecture	73
4.4.3 Implementation	76
4.5 Radix conversion	77
4.5.1 Binary to Ternary	77
4.5.2 Ternary to Binary	78
4.6 Conclusion	79
5 Discussion	81
5.1 Towards a ternary technology stack	81
5.2 Open questions	82

VIII

Contents

6 Conclusion	83
---------------------	-----------

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

A uMemristorToolbox: Open source framework to control memristors	113
B Automated synthesis of ternary logic functions in CNTFET circuits	121
C Post-Binary Robotics: Using memristors with ternary states	127
D Ternary computing; The future of IoT?	135
E High speed bi-directional binary-ternary interface with CNTFETS	143
F Ternary and mixed radix CNTFET circuit design, simulation and verification	151
G Additional material	159
G.1 Continuous-time and discrete-time signals	160
G.2 Rebuttal of Buchholz' 9 arguments	161
G.3 A radix compatible form of the CMOS power equation	162
G.4 Using the radix economy argument	164
G.5 Overhead calculation	166
G.6 Comparing baselines to 58.5% or to 63.1%	167
G.7 58.5% is an information limit, not a system limit	168
G.8 Ternary computers architectures from 2004-2022	169
G.9 Experimental 2-trit memristor results using uMemristorToolbox	171
G.10 Multi-state RRAM development platform prototype	173
G.11 Getting started with MRCS	174
G.12 MRCS Limitations	176
G.13 Ternary algebra	179
G.14 Towards a ternary standard cell library	181
G.15 Combinatorial and sequential building blocks	185
G.16 Subcomponents	194
G.17 Online radix conversion tool	197
H TNNN 2023: Ternary VLSI with CMOS using MRCS	199
I Tape-outs	203

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

X

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

LIST OF PAPERS

Article 1

S. Bos, H. Gundersen and F. Sanfilippo, "uMemristorToolbox: Open source framework to control memristors in Unity for ternary applications", *2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, Virtual Conference, Japan, 2020, pp. 212-217, doi: 10.1109/ISMVL49045.2020.000-3.

Article 2

H. N. Risto, **S. Bos** and H. Gundersen, "Automated synthesis of netlists for ternary-valued n-ary logic functions in CNTFET circuits", *2020 Proceedings of the 61st Conference on Simulation and Modelling (SIMS)*, Virtual Conference, Finland, 2020, pp. 483-485, doi: 10.3384/ecp20176483

Article 3

S. Bos, J. B. Nilsen and H. Gundersen, "Post-Binary Robotics: Using Memristors With Ternary States for Robotics Control", *2020 IEEE 8th Electronics System-Integration Technology Conference (ESTC)*, Virtual Conference, Norway, 2020, pp. 1-6, doi: 10.1109/ESTC48849.2020.9229820.

Article 4

H. Gundersen and **S.Bos**, "Ternary Computing; The future of IoT?", *2021 25th Proceedings of the Society for Design and Process Science (SDPS)*, Virtual Conference, Norway, 2021, pp. 43-47, link: sdpsnet.org/sdps/documents/sdps-2021/SDPS%202021%20Proceedings.pdf

Article 5

S.Bos, H. N. Risto and H. Gundersen, "High speed bi-directional binary-ternary interface with CNTFETS", *2021 25th Proceedings of the Society for Design and Process Science (SDPS)*, Virtual Conference, Norway, 2021, pp. 38-42, link: sdpsnet.org/sdps/documents/sdps-2021/SDPS%202021%20Proceedings.pdf

Article 6

S. Bos, H. N. Risto and H. Gundersen, "Beyond CMOS: Ternary and mixed radix CNTFET circuit design, simulation and verification", *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, Austin, TX, USA, 2022, pp. 80-85, doi: 10.1109/ISCAS48785.2022.9937259.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

List of Co-supervised Projects

1. **Halvor Nybø Risto**, "A study of CNTFET implementations for Ternary Logic and Data Radix Conversion", *Master Thesis*, USN, 2020.
2. **Julian Breivold Nilsen**, "Memristor Implementation of a Ternary Storage Circuit", *Master Thesis*, USN, 2020.
3. **Mehtab Singh Virk**, "Memristor Development Platform - Dual Source Control For Implementations of Multi-state Memristive Memory", *Master Thesis*, USN, 2022.
4. **Erika Fegri**, "Design of a Balanced Ternary Tri-directional Loadable Counter Using CNTFETs", *Master Thesis*, USN, 2022.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

List of Figures

1.1	Performance, Power and Area: 48 years of CPU innovation	4
1.2	The computer technology stack	5
1.3	The power wall and dark silicon trend	10
1.4	The memory wall	12
1.5	The memory pyramid	13
1.6	The EDA wall	14
1.7	Trade-offs to increase the radix	19
2.1	20 year IEEE Explore trend for search "ternary AND comput*"	28
3.1	The 3 regions of voltage-controlled bi-polar memristance programming . .	48
3.2	Architecture of uMemristorToolbox with memristor development platform	50
3.3	Board check experiment	51
3.4	DC experiment	52
3.5	Random write experiment	53
3.6	Multi-state programming scheme interface	53
3.7	Retention experiment	54
3.8	ADC experiment	55
3.9	ADC experiment log	55
4.1	MRCS architecture and workflows	61
4.2	User interface of the developer version of MRCS	62
4.3	The RTL-to-GDS ASIC flow from OpenLane	64
4.4	Schematic and simulation excerpt of a BCT counter in Vivado	65
4.5	The mixed radix synthesis algorithm	68
4.6	Logic transformation from truth table to CNTFET implementation	69
4.7	BCT implementation of an STI made with MRCS	71
4.8	REBEL-2 balanced ternary CPU and 9-instruction RISC-like ternary ISA .	75
4.9	382T 4-bit signed binary to 4-trit balanced ternary radix converter . . .	77
4.10	Radix converter circuits for signed binary and balanced ternary	79
G.1	Comparison of digital and analog signals	160
G.2	Measurement of nine memristance levels	171
G.3	Simulation of nine memristance levels	172
G.4	A novel multi-state RRAM development platform	173
G.5	First PCB implementation of the multi-state RRAM development platform	173
G.6	Overview of used memristors	173
G.7	Mixed Radix Circuit Synthesizer (MRCS) user experience	174
G.8	Verilog workaround for BCT with ternary d-latch	176

[Download full-text PDF](#)[Download citation](#)[Copy link](#)*List of Figures*

G.9	Heptavintimal implementation in MRCS	178
G.10	28T gated balanced ternary d-latch based on 2:1 MUX	185
G.11	46T gated balanced ternary d-latch based on NMIN	185
G.12	54T rising-edge master-slave configuration balanced ternary d-flip-flop	186
G.13	52T rising-edge master-slave configuration unbalanced ternary d-flip-flop	186
G.14	76T DDR master-slave configuration balanced ternary d-flip-flop	187
G.15	110T QDR master-slave configuration balanced ternary d-flip-flop	187
G.16	80T balanced ternary register	188
G.17	RAM-3 implementation with three d-flip-flops	188
G.18	Ternary ROM/RAM	189
G.19	110T BTA design with SUM-based CARRY	190
G.20	Balanced ternary ripple counter	191
G.21	2-trit balanced ternary ripple counter	191
G.22	1-trit synchronous balanced ternary program counter	192
G.23	4-trit synchronous balanced ternary program counter	193
G.24	MUX level 2 implementation	194
G.25	MUX level 1 implementation	194
G.26	DEMUX level 2 implementation	194
G.27	DEMUX level 1 implementation	194
G.28	binary XOR-3 implementation	194
G.29	binary HA-3 implementation	194
G.30	Conditional-STI-3 implementation	195
G.31	4-bit unsigned binary to balanced ternary radix converter	195
G.32	3-trit balanced ternary to 4-bit signed binary radix converter	196
G.33	Online radix converter tool	197

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

List of Tables

1.1	Relation between discrete radices, compactness and ambiguity	3
1.2	Relation between chapters and relevant papers	25
2.1	Radix-2 and radix-3 MAX truth tables	29
2.2	Heptavintimal (radix-27) encoding	30
4.1	Variants of binary coded balanced ternary	70
G.1	Bi-stable subset of ternary unary functions	180
G.2	Overview of useful arity-1 building blocks	182
G.3	Overview of useful arity-2 building blocks	183
G.4	Overview of useful arity-3 building blocks	184

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Nomenclature

Symbol	Explanation
3VL	Three-Valued Logic
ADC	Analog Digital Conversion
ALU	Arithmethic Logic Unit
ASIC	Application Specific Integrated Circuit
BCD	Binary Coded Decimal
BCT	Binary Coded Ternary
BTA	Balanced Ternary Adder
CMOS	Complementary Metal-Oxide Semiconductor
CNTFET	Carbon Nanotube Field-Effect Transistor
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
EDA	Electronic Design Automation
FPGA	Field-Programmable Gate Array
HRS	High Resistance State
IC	Integrated Circuit
ISA	Instruction Set Architecture
LRS	Low Resistance State
MOSFET	Metal Oxide Silicon Field Effect Transistor
MRCS	Mixed Radix Circuit Synthesizer
MVL	Multiple-Valued Logic
NTI	Negative Ternary Inverter
PCB	Printed Circuit Board
PDP	Power Delay Product
PPA(C)	Performance, Power, Area (and Cost)
PTI	Positive Ternary Inverter
PVT	Process, Voltage, Temperature
RBR	Redundant Binary Representation
REBEL	RISC-V-like Energy efficient Balanced tErnary Logic
RISC	Reduced Instruction Set Computer
RRAM	Resistive Random Access Memory
RTL	Register Transfer Level
SPICE	Simulation Program with Integrated Circuit Emphasis
SRAM	Static Random Access Memory
STI	Standard Ternary Inverter
TCB	Ternary Coded Binary

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

XX

—“At an early point in my thinking about digital computers, I looked at the effects of a change in the base of the number system. How would the structure of a computing machine depend on this choice?”

Prof. John V. Atanasoff, Inventor of the binary computer [1]

—“Computer designers have a common goal: to find a language that makes it easy to build the hardware and the compiler while maximizing performance and minimizing cost and energy”

Prof. Patterson and Prof. Hennessy, authors of Computer Organization and Design in [2]

1

Introduction

1.1 The computer alphabet

For more than 80 years digital computers read, write and “think” in zeros and ones. Remarkably, long and complex patterns of just these two symbols control computers to land a spaceship on the moon, forecast the next Aurora Borealis or provide endless digital entertainment. Computers have propelled us from the industrial age to the information age and these two symbols are at the center. In contrast, most people perform mental calculation with 10 symbols, the digits 0-9. English speakers use an alphabet of 26 symbols. Ancient Greek numerals combined the two symbol sets as each number corresponded to a letter in the alphabet. The fact that most scholars associate α (alpha) with *one* or *first* is a direct consequence of this.

The two-letter computer alphabet structures a computers lowest level language and is known as *binary*. In mathematics and computer science the number of unique symbols in a set to represent a number is called the base or radix (Latin: root) and is analog to the amount of letters in an alphabet. Binary is thus radix-2, ternary with three symbols is radix-3 and denary with its ten symbols 0-9 is radix-10. To express a large number with a set of smaller numbers, a system of transformation rules and notation is required. The ancient Greek numeral system was superseded by the the Hindu-Arabic positional numeral system as it had several advantages. It introduced symbols exclusively for numbers and

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

”10” can be interpreted as something other than a number and is the reason why modern computers should be considered symbol or computer language processors rather than arithmetic processors. Patterson and Hennessy wrote [2]:

1

1 Introduction

”No matter what the instruction set or its size — RISC-V, MIPS, ARM, x86— never forget that bit patterns have no inherent meaning. The same bit pattern may represent a signed integer, unsigned integer, floating-point number, string, instruction, and so on. In stored-program computers, it is the operation on the bit pattern that determines its meaning.”

In *Origins of Language* [3] Tomasello wrote that human language differs from other animal species in two main ways: humans use symbols and grammar. Central in human language are the various writing systems [4], a system for recording and conveying messages such as the alphabet. Man [5] claims the alphabet to be *”one of humanities greatest ideas”*. In alphabetic writing systems the smallest units for causing a contrast in meaning are called *graphemes* for symbols. For the physical representation of speech as sound patterns the smallest units are called *phonemes*. Both Pae [4] and Crystal [6] write (paraphrased) ”in a perfect regular system there is one grapheme for each phoneme”. Such a system allows compact encoding of the language while at the same time offer great expressive power needed for labelling new concepts [4]. Alphabetic writing system are recognized as the most economic and versatile of all writing systems [4]. The alphabet size of human languages varies from 11 in Rotokas to 74 letters in Khmer [6]. In modern computers graphemes are depicted as the digits 0 and 1 which correlates 1:1 to their physical representation often expressed as a range of voltage levels such as 0 V - V_{DD} for symbol 0 and V_{DD} for symbol 1. Other electrical quantities can be used as physical representation such as resistance as well as other energy domains such as mechanical, thermal and optical.

Power is the number one design constraint for designing computers [7], [8]. Interestingly, modern computers spend 1000x more energy on communication than on computation and this is increasing with every new generation of smaller chips [9], [10]. If language and its structure is so efficient for humans why did the pioneers of electronic computing Atanasoff, von Neumann and others advocate for binary? Does a richer computer alphabet, a higher radix, have the same benefits as it does for humans? If so, then uprooting this foundation of computing is a radical paradigm shift that effects all digital computers.

1.2 Motivation and scope

The question of radix, namely choosing the alphabet size, has been prevalent since the

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

radix-10 [11, p. 11]. The choice for radix-10 for addition and subtraction was based on the 10 human fingers [12]. The importance of choosing a radix can be observed in table 1.1, where the numbers 0-10 are encoded in radix-1, radix-2, radix-3, radix-8 and radix-10. The formula for encoding any positive number n in radix form is Eq. 1.1 [13]. The solution is unambiguous only for that radix.

2

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

1.3 Historical evolution of binary computing

The first landmark paper was Alan Turing's 1936 "On computable numbers" [44]. This theoretical machine used unary encoding ("tally counting") and laid the mathematical foundation for computing. In the paper Turing described *universal machines*, machines that could do more than arithmetic and are akin to general purpose symbol processors. Turing and von Neumann had met on several occasions and this paper is quoted to have inspired von Neumann [45, p. 10]. Turing also proposed and constructed practical computers, most notable the 1945 ACE [37] which was based on his 1936 paper.

The second influential paper was the Master thesis written by Claude Shannon [46] in 1937. This work laid the foundation for EDA tools describing in detail a link between Boolean algebra and circuit implementation using binary switches. For example, he showed how to synthesize complex functions such as a n -bit full adder with sum and carry signal into a electrical circuit.

The third key paper is a manuscript written by John Vincent Atanasoff [47] who through his pioneering work between 1935 and 1941 is considered by some to be the father of the modern computer [48], [49]. He explicitly investigated the role of radix for building computers [38, p. 307]:

"Considerable thought was given tot the design of a computing mechanism that would simultaneously be simple, fast and accurate. After many attempts to devise a conventional computing mechanism with these properties attention was turned to the possibility of changing the base of the numbers in which the computation is carried out. For a short time the base one-hundred was thought to have promise but a calculation of the speed of computation carried out in terms of this base showed it to be so low as to make its use out of the question. However this same calculation showed that the base that theoretically gives the highest speed of calculation is e , the natural base. But the base of numbers must be an integer, and a further calculation indicated that the bases two and three yield number systems with the same and consequently the highest speed of calculation. The choice of the base for a system of numbers to be used for mechanical calculation is a rather different question than if the numbers are to be used in mental calculation."

Together with his PhD student Clifford Berry he build the first electronic binary computer, the Atanasoff-Berry-Computer (ABC). The computer featured many engineering novelties such as electronic switching using vacuum tubes for logic and charge-based storage with capacitors for memory [38]. These two devices were designed to be inherently bi-stable as each device is capable of representing 2 stable states. With them he discovered the devices to implement radix-2 efficiently in computers. It was also very close to the theoretical

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

1 Introduction

The fourth seminal paper was a report by John von Neumann on the working details of a new computer architecture [51]. The foreword by Godfrey mentions that this paper inspired the first generation of computer engineers. Von Neumann joined John W. Mauchly's project as advisor to build the successor to the radix-10 ENIAC, the radix-2 EDVAC. Mauchly visited Atanasoff's lab during ENIAC's development and wrote to Atanasoff he was considering to implement his digital approach. Mauchly denied being inspired by it though. A court ruling ended a patent dispute, finding the ENIAC a derivative of the ABC [1], [48]. More controversially, von Neumann wrote a preliminary report based on the internal discussions of the radix-2 EDVAC computer without referencing Mauchly and others. In the same year he published a report with Goldstine and Burke [52] discussing the role of binary. The main argument favoring binary he gave was based on pragmatics [51]:

"Thus, whether the tubes are used as gates or as triggers, the all-or-none, two equilibrium arrangement are the simplest ones. Since these tube arrangements are to handle numbers by means of their digits, it is natural to use a system of arithmetic in which the digits are also two-valued. This suggest the use of a binary system. The analogs of human neurons are equally all-or-none elements."

It is worth noting that this last sentence was based on the 1943 paper by McCulloch-Pitts [53]. This paper was the first computational model of the human brain and was proven quite early to be far too simplistic and inaccurate [19].

The fifth paper considered to be influential for the mass adoption of radix-2 in modern computers is by Werner Buchholz [12]. The 1955 "Fingers or Fist" paper is one of the few academic works that focussed on radix comparison. Buchholz worked for IBM and collaborated with Gerrit Blaauw on computer architectures [54]. Blaauw is the "inventor" of the 8-bit byte [55] and lead architect of the IBM/360, one of the most successful mass-produced computers. In the Fingers (radix-10) or Fists (radix-2) paper, Buchholz wrote that radix-2 was superior to radix-10 for nine reasons. Buchholz cited von Neumann's EDVAC paper for several of the reasons. Important is that for many of the arguments, radix-10 was implicitly assumed to be implemented with bi-stable devices (and encodings such as binary coded decimals, bi-quinary, etc) as this was the most efficient implementation. A brief analysis of the 9 arguments can be found in the **Appendix G.2**.

Lastly, work with less focus on structured radix comparison from this era was "Arithmetic Operation in Digital Computers" from 1955 by Richards discussing radix-2, radix-3 and radix-10. They mention that the ternary system was seriously considered. An important quote from that work is [56, p. 15]:

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

with the difficulties in adapting the system to applications where the decimal

8

1.4 Moore's curse: 3 scaling walls

system is already entrenched, it appears that the disadvantages of the ternary system with positive and negative coefficients substantially outweigh the advantages.”

Shannon's 1950 "A Symmetrical Notation for Numbers" [23] discussed the properties of symmetrical radices around zero (such as radix-3, radix-5, etc) and mentioned that some of the benefits disappear when for instance radix-3 is used with asymmetric arithmetic. This topic will be relevant in Chapter 2 on balanced vs unbalanced ternary. The third honorable mention on radix comparison is the 1950 work by the Engineering Research Associates Staff who briefly mention symmetrical radix-3 (balanced ternary) adder designs [57, p. 287] but do not divulge in a comparison to binary. The first electrical ternary computer, the 1958 Russian Setun [26]–[28] actually used binary coded ternary (2 bits for 1 trit) [58] and will be discussed in Chapter 2. This computer had some commercial success, unique features and piqued interest from the USA [58] but could not influence the path towards binary dominance. Radix-2 was the best option because the radix implementation was done with bi-stable binary logic and memory devices. Implementing other radices with them would make them economically inefficient.

Readers that are interested in historical events after the origins of binary computers are referred to the IEEE Annals of the History of Computing. Central is the focus of miniaturization of bi-stable transistors after the invention of integrated circuits. This became the cornerstone for binary computers.

1.4 Moore's curse: 3 scaling walls

Moore's law has inspired continuous device-centric scaling for over 50 years and counting through a multi-disciplinary effort involving both academia and industry. Revenue in the global semiconductor industry has grown to 700 billion USD in 2023 [59]. The writing on the wall started after Dennard scaling [18] stopped around 2005. Transistors scaled physically afterwards, but other properties such as power consumption did not. The relentless exponential growth by doubling transistor density every two year is unusual compared to other industries [60]. Like any exponential growth curve it is bound to end. Perhaps more importantly, the continued focus on the area metric masks technical debt. Moore's law has become Moore's curse [60]. Performance and power show marginal growth of 3% for nearly 20 years (see Fig. 1.1) and [2, p. 44]. The 2022 Industry Roadmap for Devices and Systems (IRDS), a leading set of frequently updated white papers [61],

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

to explore higher radices [62, p. 4]. Three categories of challenges relevant for this thesis are highlighted: power consumption, memory access and EDA tooling and workflows. All three challenges can be considered corollaries of Moore's Law.

9

1 Introduction

1.4.1 Power wall

Figure 1.3: (Left) The power wall, a post-Dennard power density trend. (Right) Dark silicon: projected transistor activity over technology nodes. Both plots adapted from [63]

The *power wall* [64] refers to the increasing problem of power delivery and heat dissipation due increased transistor density (see Fig. 1.3). The problem is not limited to billions of fast switching transistors. Each transistor is wired up from its gate, source and drain terminals. The increasingly lengthier interconnects of these transistors also dissipate heat during switching. The power consumption equation is discussed in **Appendix G.3** as it is slightly more complex for higher radix signals. Power is considered the number one design constraint [8] and limits both low-power and high performance computers [64]. The effect of the power wall visible in Fig. 1.1 shows limited single threaded performance increase after 2005. Dennard's paper [18] observed two major scaling rules. The first was related to scaling interconnects mentioning that "*Scaled interconnects provide roughly constant RC delays because the reduction in line capacitance is offset by an increase in line resistance*" [65]. This means that as interconnects became smaller no speed was gained. A detailed analysis of the interconnect problem or "interconnect wall" in relation to scaling can be found in [66]. The second scaling rule was about transistor scaling, a formula to improve

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Unfortunately the second scaling rule is no longer feasible which marks the end of Dennard scaling. The rule to double transistor density required a reduction of transistor dimensions by a factor of 0.7 and keep power density constant by reducing supply voltage by the same factor. To reduce the supply voltage while maintaining a good I_{ON}/I_{OFF} ratio at room

10

1.4 Moore's curse: 3 scaling walls

temperature the voltage threshold needs to scale down proportionally. For 30 years this could be done by scaling the gate oxide thickness. As gate oxide thickness approached just 5 silicon atomic layers in the early 2000's, transistors couldn't switch off properly anymore resulting in constant leakage current [65]. Multiplied with billions of transistors this small leakage became significant - even when doing no computation. This phenomenon is called direct tunneling. Other issues are known as short-channel effects (SCE) [67, p. 496-504] [68] and play an increasing role in sub 100 nm technology nodes [69].

Power density is defined as $D = P_{device}/Area_{device}$. With the ongoing scaling of $Area_{device}$ and without Dennard scaling of P_{device} , power density keeps increasing every technology node (see Fig. 1.3). Transistors operating at 1.5 THz f_{max} have been demonstrated [70]. In practise a fraction of that maximum frequency can be used for switching activity. Operating at lower frequencies to curb power is called dim silicon. The phenomena of disabling transistors is called dark silicon [63] and has been increasing every technology node (Fig. 1.3). Markov also classifies grey silicon, which are additional non-functional but power consuming structures such as repeater gates that are needed because the interconnects have become too short [7]. Repeater insertion, both the amount and placement, is an increasing problem with every node [71]. On 130nm interconnects including repeaters can form 50% of the dynamic power consumption and leakage power [71]. Repeaters are critical for minimizing clock skew in the clock tree network (CTN) and signal integrity. Ideally these repeaters are placed in higher metal layers where there is less interconnect congestion and would reduce the amount and placement problems [72]. Currently, repeater gates share the same physical layer as the other logic gates. Device utilization is considered a main challenge in the 2022 IRDS roadmap [61, p. 3]. If power consumption goes unchecked temporary or permanent thermal related defects arise. Defects include lifetime, performance and reliability.

Various approaches have been discussed in literature [8], [64], [73], [74] to reduce power consumption and curb heat dissipation in hardware. At the system level the most common strategy to mitigate heat dissipation is to use active cooling solutions. Good heat transfer reduces leakage current as the thermal noise floor is lowered [75]. A cool CPU allows higher switching activity α and F_{clk} because electron mobility is negatively affected by temperature. Active cooling require power drawn from the system and can be significant (for instance 40% in supercomputers [75, p. 10]). Recent innovations in the field include

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

1.4.2 Memory wall

The *memory wall* [77]–[79] is the latency and bandwidth gap between on-chip data processing and off-chip data retrieval from dynamic RAM (DRAM), see Fig. 1.4. It is also

11

1 Introduction

Figure 1.4: The memory wall. The gap between retrieving data and operating on it. Adapted from [78]

called the von Neumann-bottleneck [80]. The gap stabilized as frequency couldn't scale after Dennard scaling stopped. However, in technology nodes after 10 nm Moore's law seems to saturate for both DRAM [81] and static RAM (SRAM) [82] scaling meaning that the gap might widen further again. For DRAM 50% more processing steps are needed to move from technology node $2X$ to $1Z$ and include costly double and quadruple patterning steps [81]. Both logic cells and memory cells suffer from lower supply voltages and increased interconnect resistance as a result of scaling. Worse, for memory cells the decrease in supply voltage and increase in interconnect resistance degrades SRAM and DRAM performance [81], [83] making it harder to scale voltage further. The different technical requirements for logical and memory is part of the reason why DRAM is made on much older technology nodes [81, p. 1386]: "*The low leakage is required to prevent the discharging of the capacitor, and the high ON-current is expected to write the data in a short time*".

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

This research paper [52] and consists of 6 components, datapath, control, input, output and memory. The datapath and control components are together called the processor[2]. Memory physically located on the same die as the processor has the shortest path to the processor and is called *cache* or on-chip memory. This type of memory shares the die area with digital logic blocks and analog blocks. The area SRAM occupies depend on the product but even processor's from 2004 like the Intel Itanium 2 6M can occupy 50% [84] with L2/L3 cache. In terms of power SRAM can consume 25-50% of the total power [78]. The further the memory is from the processor the more power and time it takes to retrieve or access the data, see the memory pyramid in Fig. 1.5.

12

1.4 Moore's curse: 3 scaling walls

Figure 1.5: The memory hierarchy and latency pyramid. Adapted from [86]

Since 1975 on-chip memory is high density SRAM while the nearest off-chip memory is DRAM [78]. SRAM is typically made with 6 transistors, a 6T-SRAM cell [83]. DRAM is made with 1 transistor and a capacitor, a 1T1C-DRAM cell [81]. To improve latency both SRAM and DRAM cells use pre-charge circuits [85] such that the voltage swings are halved or reduced. The average memory access time, the latency, is data dependent as data in cache doesn't require hundreds of clock cycles to fetch from DRAM [10]. A 32-bits memory load also costs 1300x more energy then a 32-bits ALU operation on a 45nm process [10], a significant increase from 260x at 130nm. Electron transport is more

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

The stored-program concept by von Neumann [51] is the architectural reason for the bottleneck between memory and the processor. Both the Harvard and von Neumann architectures define that computers cannot perform operations directly on memory. Using a memory hierarchy like in Fig. 1.5, data is stored and retrieved from memory and moved to a central processing unit. All conventional computers use this architecture and thus implement some sort of fetch-execute-writeback mechanism. New memory paradigms such as compute-in-memory (CIM) or processing-in-memory (PIM) using novel memory cells such as RRAM and a higher radix can break this paradigm [80], [87].

13

1 Introduction

1.4.3 EDA wall

Figure 1.6: The EDA wall. The exponential increase of chip design costs with transistor density. From [88] (original data: IBS)

The exponential increase in electronic design automation (EDA) related cost every technology node [88]–[91] can be considered another wall, the *EDA-wall* and is visible in Fig. 1.6. Moore's exponential pace of scaling enables more transistors for the same design space pressuring EDA manufacturers to scale design, simulation and verification time at the same pace to keep cost equal [92, p. 219]. The average amount of design rule checking

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

circuit (IC) design and layout engineers. The first microprocessor, the Intel 4004 from 1971 had just 2300 transistors. The logic design, circuit design, layout design and verification was mostly done by a single person, Federico Faggin [95]. Development of silicon products in those days were done without computer aided design (CAD) tools, a synonym for EDA tools [92], [96]. When complexity grew to several thousands transistors in the mid 1970's, EDA tools such as SPICE circuit simulators and place and route (PnR) tools became a necessity. With an abundance of tools came also the need for deeper integration and automated workflows. A comprehensive historical overview about the period 1964-2002 can be found in [96] . Despite nearly 70 years of academic interest in higher radix computing few MVL EDA tools exist to accommodate design and verification of MVL circuits [97], [98]. These tools could aid in scaling the EDA wall for both binary and MVL circuits as some Boolean logic problems are better solved in the MVL domain [97], [99].

14

1.5 The fundamental limits of computing

Transistor density and verification is not the only reason why the EDA wall grows exponentially. The enormous innovation to keep up with Moore's law forces the industry to keep developing new EDA functionality, tooling and workflows. Advances in materials science (such as high-k dielectrics, memristive materials), device evolution (such as planar MOSFET, finFET, GAAFET, CNTFET) and architectures (such as interconnect stackups, superscaler, in-memory compute) all need software to be supported. In 2018 DARPA launced a 100 million open source investment to drastically curb costs of modern silicon and control the EDA wall. This initiative had lead to OpenRoad [100], Openlane [101] and SiliconCompiler [102] which aim to provide complete or partial RTL to GDS flows. Openlane is used in this thesis work. These flows rely on open process design kits (PDK) which are considered the crown jewels of a foundry. Examples of open source PDK's are Skywater foundry's 90nm and 130nm, GlobalFoundries 180nm and ASAP7, a 7nm FINFET PDK [103].

Another popular measure to reduce cost and complexity is reusing battle-tested components, so called intellectual property (IP) blocks. This is a very similar to the reusable component principle in software engineering. According to [91] a modern System-on-Chip (SoC) uses on average over 175 IP blocks with just 20% of the design being custom. Many commercial chips are no longer designed as monolithic. Rather a multi-process or *chiplet* approach is used, merging several dies on a single substrate to form a complete SoC. Gordon Moore already predicted the cost benefits of this strategy in his 1965 paper [17].

The integration of artificial intelligence (AI) might also reduce cost and complexity. The launch of AI accelerator chips to the market is increasing astronomically [104] and influences many industries. The EDA industry has an interesting dependency dynamic with AI accelerators. EDA functionality is being more and more integrated with AI such as

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

is that AI for verification (and validation) is still lacking, often because the test patterns depend on the design.

1.5 The fundamental limits of computing

The fundamental limits of computing has historically excited computer researchers [7], [107]–[112]. The discussion if Moore’s law has already ended has devolved into semantics. Density scaling is certainly ongoing [15], but the same source shows that transistor costs is not following the same exponential scaling. Renowned chip designer Jim Keller shows that pushing Moore’s law has never been a single technology development but rather a series of S-curves [113]. It undeniable though that several parameters depicted in Fig. 1.1 have not seen exponential growth for decades. The availability of more transistors

15

1 Introduction

made up for the inefficiency of the whole system. This is a similar to the abstraction in higher level programming languages. Simplifying programming with natural language increases application development but at the cost of application performance and power consumption [114].

Investigating the influence of radix on computational limits requires an overview of fundamental principles that govern computing. A comprehensive overview is published by Markov [7] and includes several physical, mathematical and practical limits in domains such as material, devices, circuits and software. From these Shannon’s limit and Landauer’s limit stand out for inspection against the practical limits since it uses radix-2 in its formulation. In addition the mathematical theorem on radix economy is added and Rent’s rule, a complexity heuristic that is directly effected by the radix.

1.5.1 Shannon’s limit

In 1948 Claude Shannon published ”A Mathematical Theory of Communication” in which he extended the theories from Nyquist and Hartley [115]. He was motivated by the various methods that exchanged bandwidth for signal-to-noise ratio (SNR) such as Pulse Code Modulation (PCM). In the paper he proposes a fundamental theory of communication by modelling the effect of noise in the channel and exploiting statistical structure of the source message. His error-free noisy-channel coding theorem also known as Shannon’s limit introduces the concept of coding to approach this channel capacity limit. The theory predicts an upper bound, the channel limit, which is measured in a radix of choice such as bits/sec. The channel limit is a mathematical limit on how much information

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

of the theorem, known as the Shannon-Hartley theorem, to analog (continuous signal) communication through interconnects. Interested readers are encouraged to read the complete presentation of the theorem which includes the nature of information (entropy) and discrete and continuous noisy and noiseless channels in [115]. A less mathematical presentation of the paper with many applications beyond Shannon's work is written by John R. Pierce [116].

The equation of the Shannon noisy-channel theorem for continuous channels is shown in Eq. 1.2 [115, p. 43].

$$C = W * \log\left(\frac{P+N}{N}\right) \quad (1.2)$$

Where C is the channel capacity, W is the bandwidth (in Hz) and log the logarithm in some radix of choice. P is the average signal power and N is average noise power.

16

1.5 The fundamental limits of computing

When additive white Gaussian noise (AWGN) is assumed [116, p. 173] and information is measured in radix-2 (bits) then the equation becomes the Shannon-Hartley theorem Eq. 1.3 [115, p. 45]

$$C = W * \log_2\left(1 + \frac{P}{N}\right), \text{ in bits/sec} \quad (1.3)$$

For given parameters bandwidth W and SNR P/N a higher radix than 2 can be chosen. For example, choose $W = 1.585$ and $P/N = 1$. Then $C = 1.585$ bits /sec. If radix-3 is chosen with the same parameters, then $C \approx 1$ trit /sec. Measuring in a higher radix does not mean that additional information is gained, only that the information is now more compactly encoded. The example shows that information is encoded in 1 symbol in ternary versus 1.585 symbols in binary. Equation 1.3 also shows two obvious limits, W and SNR. Bandwidth is discussed in the next subsection. At or below the noise limit, where SNR approaches 0 or is below it, binary is the only option [116, p. 176]. In that scenario one or multiple binary symbols needs to be send to be decoded as a single noise-free bit.

An alternative form of Eq. 1.3 is derived from Hartley and Nyquist's work by expressing channel capacity as $n * \log * r$, the maximum number of independent pulses n that can be transmitted per second using r levels (such as voltage amplitude levels, the radix). Using Nyquist's theorem n can be replaced by $2 * W$ (twice the bandwidth). The equation can be rewritten such that an expression for the channel capacity is obtained with both W and radix, measured in bits (\log_2 , Eq. 1.4) or in trits (\log_3 , Eq. 1.5) :

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

$$C = W * \log_3(r^2), \text{ in trits/sec} \quad (1.5)$$

Where r is expressed as $\sqrt{1 + \frac{P}{N}}$

In these forms it can be seen that to send for example ternary signals noise free for some capacity C, a minimum SNR is needed of $r = 3 = \sqrt{1 + \frac{P}{N}} = 9$. This means that SNR ≥ 8 to be received noise-free. At this lower limit, the average transmission rate requires block coding [116, p. 177]. Transmission without block coding is only possible if the transmitter uses a brief but powerful pulse [116, p. 177]. This gives a good argument for pulse-based signaling architectures in some scenario's such as (ternary) spiking neural networks [117].

Shannon's error-free noisy-channel coding theorem shows through mathematical rigor [115] that binary is fundamentally the simplest and in some extreme cases the only communication signal to obtain the channel limit. In most practical scenarios higher radices

17

1 Introduction

can be used in such a way that the same channel capacity or higher is obtained. This will be discussed in more detail in Chapter 2. Approaching the limit to an arbitrary low error-rate has both hardware and software consequences.

Approaching the limit with software

The noisy-channel coding theorem introduces the principles now found in data compression theory [118, p. 6] using the concept of adding and removing redundancy to increase information entropy (the average amount of information per symbol) and robustness to noise. This conclusion is best summarized in a quote by Pierce [116, p. 164]:

"Indeed, the whole problem of efficient error-free communication turns out to be that of removing from messages the somewhat inefficient redundancy which they have and then adding redundancy of the right sort in order to allow correction of errors made in transmission."

In compression theory both the modelling and coding phase depend on the radix [118, p. 6]. In the modelling phase a data source is modelled. In some situations modelling with a discrete symbol alphabet in mind can be beneficial, for example a data source model with an alphabet of 27 symbols can be efficiently encoded in radix-3 with 3 trits/symbol in the coding phase. With a richer alphabet more patterns can be made such that they

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Approaching the limit with hardware

The Shannon-Hartley equation shown in Eq. 1.3 models a continuous signal communication channel. In the context of computers such a signal is an analog electrical signal and flows through the dense and lengthy network of interconnects. The channel capacity C is the amount of bits/sec that can flow through a single interconnect. The interconnects are bandwidth-limited, not power limited as the SNR is ≥ 1 . The SNR is such that error likelihood is extremely small during transmission. For example the bit error rate (BER) of the USB 4.2 spec using ternary signalling is $1E^{-19}$ (coded) and $1E^{-8}$ (uncoded) or 1 error in 10^{19} bits send [119]. For high-speed communication block coded signals are common, but for memory cells and logic gates uncoded signals are used.

The bandwidth for computing is the highest frequency possible measured in Hertz such as the clock signal. For interconnects the properties of the conductive material and surrounding insulator attenuate the signal increasingly with higher frequencies. The effect is that a square input pulse becomes a flat, spread out signal [116, p. 26-38]. The degraded signals effects are called attenuated peak-to-peak voltage swings and inter-symbol interference (ISI) [120]. To compensate for these effects higher power is needed which makes higher frequencies after 25GHz energy inefficient in CMOS circuits [121]. The practical

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

4.3 Mixed radix synthesis engine

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure 4.6: Logic transformation from truth table to CNTFET implementation using MRCS's synthesis algorithm described in Fig. 4.5 and [Paper B](#)

4.3.4 Binary coded ternary RTL

Ternary synthesis with ternary signals using CNTFET has not been demonstrated outside simulation. Ternary synthesis with BCT signals on the other hand is very feasible and is demonstrated to occasionally outperform binary [188], [189]. BCT allows experimenting with ternary logic design and verification with modern CMOS. Ternary logic with CMOS has been proven feasible as early as 1974 by Mouftah and Jordan [174]. In that paper a ternary And-Or-Invert (AOI) 6T2R CMOS circuit was presented. Replacing the 2R with 2T would make the 8T circuit competitive in transistor count to a standard 6T binary AOI CMOS circuit. The 5 functions the ternary AOI can perform are STI, PTI, NTI, NMAX and NMIN and would make a great standard cell.

69

4 Mixed radix EDA for ternary computers

In literature the term BCT is sometimes called *bit pairing* [263, p. 411] or *redundant binary representation* (RBR). Two bit for one trit implementations result in one degree of freedom. This has resulted in non-standard encodings for both unbalanced and balanced ternary (see Table 4.1).

Table 4.1: Variants of binary coded balanced ternary

Bit 1	Bit 0	BSD-SV	BSD-SUM	BSD-PN	BSD-PNX
0	0	0	-1	0	x (illegal)
0	1	1	0	-1	-1
1	0	0	0	1	1
1	1	-1	1	0	0

Following the naming scheme used in [188], Binary Signed Digit Positive-Negative-Exclusive (BSD-PNX) is proposed with 0b00 as illegal state for five reasons:

- Only one bit needs to change regardless of transition, reducing power consumption.
- The fourth state is an illegal/faulty/uninitialized state. Using it as a redundant logic state would invalidate that only one bit needs to change. It also removes

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- With $BSD-T$ (Δ) encoding the heavily used standard ternary inverter (SIT) becomes a simple cross wiring binary implementation, possibly with a buffer cell (see Fig. 4.7).

- Implementation with a single differential opamp allows easy conversion to ternary signals.
- Literature [188] seems to indicate that BSD-PN coding has arithmetic advantages resulting in reduced power consumption and increased performance.

The BCT verilog syntax is rather verbose as no ternary constructs can be used. Brayton and Khatri [283] created a verilog extension for MVL that might improve this. A recent specification of TernaryVerilog was also found, but the status is unknown and the source is not in the public domain [284]. Without ternary-oriented verilog BCT truth tables explode in size, especially for 3-ary truth tables such as the 2:1 ternary MUX with index $0tZD0DDPPP$. The process of writing BCT verilog in MRCS is automated and is visible in Fig. 4.7. Special attention should be paid to the limitations written in **Appendix G.12** regarding the binary or BCT verilog conversion of sequential logic such as latches.

70

4.4 REBEL-2 Balanced Ternary CPU

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure 4.7: Binary Coated Ternary (BCT) implementation of a standard ternary inverter (SI) in MRCS. For unbalanced ternary just add 1 to the balanced ternary numbers. **Top.** From MRCS component to MRCS generated verilog. **Bottom.** From ternary truth table to two binary truth tables that can be implemented with CMOS

4.4 REBEL-2 Balanced Ternary CPU

In this section REBEL-2 is presented, a novel 2-trit balanced ternary CPU made with MRCS (see Fig. 8). REBEL-2 is an acronym for RISC-V-like Energy efficient Balanced tErnary Logic CPU. The postfix denotes the 2-trit wide memory address bus. A new ternary computer should inherit the same spirit as Brousentsov when he designed the last ternary computer in 1970, the improved Setun [28]. Brousentsov's ternary principle states that a ternary computer must have [197]:

- Ternary logic
- Ternary memory
- Balanced ternary encoding

71

4 Mixed radix EDA for ternary computers

With MRCS a ternary computer can be designed that implements these principles using the building blocks shown in **Appendix G.14 and G.15**. The designer can be oblivious to the hardware implementation and use only ternary logic gates, memory and pins.

4.4.1 Motivation

A binary CNTFET CPU has been made as a proof of concept by Shulaker et al. [285] in 2013 and at a commercial foundry by Hills et al. [29] in 2019. The discovery of multi-threshold CNTFET for ternary logic was made by Raychowdhury and Roy [30] in 2005. They tuned V_{th} by changing the diameter of the nanotube while Wang et al. [286] in 2013 achieved the same effect with doping. No literature was found demonstrating a ternary CNTFET CPU outside simulations. Related work like Kam et al. (2022) [187] discuss a novel balanced ternary CPU design with ternary assembly. Their instruction set architecture (ISA) occasionally generates larger assemblies (for instance the BLT instruction expands to 3 instructions), use only two operands and does not fully exploit the benefits balanced ternary offers to reduce the instruction set even further (such as three-way branching). The paper demonstrated great benefits in various benchmarks compared to binary RISC-V designs. More recently Gadgil et al. demonstrated a ternary CNTFET

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

The REBEL-2 is the first modern ternary computer designed with the unique qualities of radix-3 in mind. It is a Harvard-style CPU with instructions in ROM and data in RAM. Compared to von Neumann-style CPU's Harvard-style design requires more area as instructions and data are not stored in the same physical memory. The benefit is that both ROM and RAM can be accessed in the same clock cycle. Arguably, the design is simpler despite additional bus interconnects and control logic. The small memory address bus and simplicity-oriented design make REBEL-2 more suitable for educational purposes than real-world applications. The architecture is heavily inspired by the book "Computer Organization and Design - RISC-V edition" by Patterson & Hennessy [2]. Reduced Instruction Set Computer (RISC) is a computer architecture that in essence shifts complexity from hardware to software allowing computer designers to optimize hardware at the cost of writing more software. Through software abstraction layers this burden is mostly hidden from the programmer. RISC uses elementary instructions while Complex Instruction Set Computer (CISC) uses higher level instructions that are decoded into simpler instructions on the hardware. Examples of RISC architectures are ARM, Atmel AVR, MIPS, SPARC, PowerPC, RISC-V. The traditional PC market uses x86, x86-64 CISC architectures for server and workstations. According to Patterson [288] CISC products have been reduced to a one percent market share while RISC make up the rest.

The REBEL-2 architecture targets low-power applications as this property is often associated with RISC architectures. The high level requirements were:

72

4.4 REBEL-2 Balanced Ternary CPU

- RISC-V-like implementation based on RV32I
- Native ternary instructions (such as ADD/SUB adder, 3-way branching and compare)
- 2-trit registers to be non-trivial gates
- No external memory with Harvard style separation of program and data
- Single cycle instructions
- Reading at rising edge, writing at falling edge
- Compact, uniform instruction format

4.4.2 Balanced Ternary Instruction Set Architecture

Central in the design of a CPU is the organisation of the instructions, the instruction set

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

includes the functionality of a CPU, with each instruction being implemented with one or more electrical data paths. The implementation of these paths determines the actual functionality and performance. Traditionally instruction sets have had little innovation and were considered a stable contract between the software and hardware domain. Few companies developed ISA's (examples include Intel, AMD, ARM). The ISA's were licensed and often non-customizable. This changed with the arrival of RISC-V in 2015 which allowed customization of the ISA and usage royalty free. Ternary computers require new ISA's or extensions to existing ISA's since their capabilities are not well captured in the currently available sets. For example, instruction formats can be more dense with 3-valued fields and more simplified without unsigned instructions. Smaller ISA's lead to simpler CPU designs. REBEL-2 has just nine instructions. Scaling this architecture to a wider memory address bus, such as a tryte¹ is possible. This would necessitate a redesign of the ISA since opcodes and function fields require much less scaling compared to the operand fields.

The memory address bus is 2-trit wide, resulting in the limited addressing of only nine 2-trit registers. Each register has 2-trit storage capacity. The most-significant-trit (MST) is the left-most trit. To fully use the nine addresses 10-trit wide instructions are used with four 2-trit operands such that most of the 40 RV32I instructions can be represented in the ISA. The opcodes are also 2-trit fields. Several instructions that do not use the

¹There is currently no standard as to how large a tryte, the equivalent of a byte, should be in ternary. A 5-trit tryte would have slightly less resolution than the 8-bit byte (243 vs 256 states) but the a 6-trit tryte would have almost 3 times more (729 vs 256) states. A 9-trit tryte is the closest to the 8-bit byte for trit-wise operations such as bit-masking.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

4 Mixed radix EDA for ternary computers

"Register Destination 2" (Rd2) field have an additional 2-trit function field to allow more functionality. Example functionality include selecting the scope between trit-wise (1-trit) or word-wide (2-trit) operation for some logic functions such as COMP. Novel functionality includes two store operations in one cycle and selection of shift left, right or cycling operation. The 32-bit RV32I base instruction set has 40 instructions [289, p. 130] of which 38 are essential [289, p. 13]. The REBEL-2 ISA has full word load and store operations such that instructions like load half word, load byte or load upper immediate can be ignored. Unsigned instructions can be ignored as only signed instruction can be optimally used with balanced ternary. Branch instruction like BEQ and BLT can be compressed to just one instruction using a ternary comparator and delegating operands ordering to the compiler. Using the function field several instructions can be merged such as AUIPC/JAL/JALR, SLLi/SRLi/SRAi/SLAi and ADD/SUB. In total 11/38 RV32i

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

shift and logic instructions which can be resolved with 2 or more instructions. The only instruction that cannot be resolved is ECALL/EBREAK to transfer control to a debugger/Operating System from an interrupt. All missing instructions could technically be fitted in unused function fields but it would obfuscate the ISA tremendously. For example, the MIMA instruction could be merged with the MUDI and XOR instruction since the format is identical and the 2-trit operand plus 2-trit functional field are large enough.

4.4.3 Implementation

Validation and performance benchmarks with comparison to binary and tapeout of the whole CPU is planned. Several key building blocks of the REBEL-2 CPU have been simulated, tested on FPGA and submitted for tape-out including the multi-trit adder, multiplier, multiplexer, RAM/ROM and program counter. They can be found in **Appendix G.15**.

The 2-trit balanced ternary ALU has eight 2-ary functions: ADD/SUB, STI, COMP, MIN, MAX, SHIFT, MULTIPLY and DIVIDE. The ninth function could be the ternary XOR function [290] such that the entire RV32i set is covered. Most of the logic functions operate on whole words but some (such as COMP) allow trit-wise operation. Demonstration video's of a 1-trit balanced ternary ALU with 5 functions and a 2-trit balanced ternary calculator [291] can be found in [292], [293].

4.5 Radix conversion

In Chapter 1.3 the history of binary computers was briefly reviewed. Computers such as the ENIAC, UNIVAC I and IBM 650 were radix-10. At the implementation level radix-10 numbers were actually made with bi-stable devices and BCD (or similar) encodings [12]. Switching to radix-2 at the architecture level meant costly radix conversion to allow human readable input and output (I/O). It is interesting to note that even today, BCD arithmetic is present in the Intel 64 and IA-32 ISA [294]. Example instructions are *Load Binary Coded Decimal* (FBLD), *Store BCD Integer and Pop* (FBSTP) and ASCII Adjust

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

hardware accelerated radix conversion method by Parikh [295] from 1971 is still relevant for general software radix conversion. Online software radix conversion with a custom developed tool is discussed **Appendix G.17**.

All modern digital electronics use binary logic. Ignoring billions of existing chips when pivoting from binary to ternary would result in an unprecedented amount of e-waste and is against UN Sustainable Development Goal 12: Responsible production and consumption. Rather, the chiplet approach discussed in Chapter 1 should be considered with either dedicated radix conversion chips or integrated conversion circuitry in ternary chips. In this section the results from **Paper E** on radix conversion are discussed which are made with MRCS from **Paper F**. Shown is how signed binary to balanced ternary conversion and the inverse is possible from theory to implementation with CNTFET and CMOS. The BCT to unary coded ternary and inverse radix converter chip is detailed in [296]). The signed binary to balanced ternary radix converter and inverse radix converter chip is discussed in **Paper E** and open sourced in [266].

4.5.1 Binary to Ternary

Figure 4.9: A 382T 4-bit 2's complement signed binary to 4-trit balanced ternary radix converter. This design reuses the 170T proposed design from **Paper E**. Components can be found in Appendix G.16

4 Mixed radix EDA for ternary computers

There is surprisingly little written on binary to ternary radix conversion in hardware (see a small survey in [282]). In particular radix conversion with signed/balanced arithmetic is rare. In **Paper E** the method by Li et al. [297] was extended to 15 bits and 10 trits. In addition the inverse matrices, balanced ternary to 2's complement signed binary, were . Our inverse matrices are also extendable and have a remarkable property: generating output in 2's complement regardless of the bit width comes straight from the matrix. The original paper by Li uses an implementation with SQUID gates for an unsigned binary to balanced ternary converter. In **Paper E** an implementation is given with CNTFET gates

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

replace each term with a balanced ternary full adder (B'TA). The other approach optimizes each BTA to the essential circuitry, for example when a carry signal is not needed. The full procedure is described in **Paper E**. The proposed 4-bit unsigned to 4-trit balanced ternary radix converter (Fig. G.31 in **Appendix G.16**) had a PDP (worst measured delay * average power consumption) of 7.824e-16 joule. The state-of-the-art for 4-bit unbalanced binary to unbalanced ternary using CNTFET [298] had a PDP of 3.024e-15 joule, an improvement of 74%. Both implementations have no capacitive load connected. The result is achieved despite it being an unfair comparison. Balanced ternary needs additional transistors for positive and negative carry signals while unbalanced ternary only needs a positive carry signal. For feature parity unbalanced ternary needs to add additional circuitry to implement 3's complement to handle negative numbers. **Paper E** reports that 176T are needed but the latest version of the synthesis engine in MRCS optimized it to 170T.

Paper E also discusses reusing the the proposed converter to make a 2's complement signed binary to balanced ternary converter by adding additional circuitry and is shown in Fig. 4.9. The implementation of the additional circuits in Fig. 4.9 can be found in **Appendix G.16**. An improved version of this radix converter would use a smaller unsigned binary to balanced ternary converter since the binary input range of -8 to +7 can be achieved with 3 trits instead of 4. Three trits have a range of -13 to +13. The overhead mentioned in **Appendix G.5** is large as even with 3 trits half of the states are unused.

4.5.2 Ternary to Binary

In **Paper E** the inverse matrices are also provided but an implementation was not included. A novel 139T 3-trit balanced ternary to 4-bit signed binary radix converter implementation made with MRCS is shown in Fig. G.32 of **Appendix G.16**. The method to construct this implementation is similar to the converter shown in Fig. 4.9 except now table 5 from **Paper E** is used.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure 4.10: The radix converter circuits from Fig. 4.9 of **Paper E** and Fig. G.32 of **Appendix G.16** used in two configurations. The top configuration is used for signed binary chips and the bottom for balanced ternary chips.

The radix converter circuits in Fig. 4.9 and Fig. G.32 are simple IP blocks that allow interfacing binary logic circuits with ternary logic circuits. As a final test, both circuits have been chained back-to-back (Fig. 4.10). The two possible configurations are useful for common scenario's where it is desired to either integrate binary chips in a ternary system or integrate ternary chips in a binary system. The back-to-back configuration has been simulated in Vivado, verified on Basys-3 FPGA and submitted for tape-out with TinyTapeout 3.5 [266]. A simulated die shot can be found in **Appendix I**. All design, test bench and constraint files for the Basys-3 FPGA are open sourced and can be found in [266].

4.6 Conclusion

In this chapter **Papers B, E and F** were discussed. Central was MRCS, the first online ternary synthesis tool to design and verify ternary logic chips. The verilog generated for BCT signals allow experimenting with ternary logic using modern binary CMOS technology. The verilog can be deployed on FPGA's, work with simulations tools like Vivado and can be hardened for tape-out as ASIC with OpenLane. The HSPICE netlist generated for ternary signals allows mixed signal simulation of ternary circuits based on emerging multi-threshold CNTFET.

4 Mixed radix EDA for ternary computers

Several verified MRCS designs have been discussed in this chapter. A comprehensive overview of standard binary and ternary building blocks is presented for both combinatorial and sequential logic in **Appendix G.14 and G.15**. To interface with binary chips, binary to ternary and ternary to binary radix conversion circuits are presented. REBEL-2,

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

The online EDA tool lays the foundation for a larger vision to design ternary VLSI and mixed radix VLSI with CNTFET, RRAM and CMOS. Future versions of MRCS will explore integration with ALIGN for analog layout [299] and integration with an RRAM compiler [300]. This enables integrating the multi-state RRAM controller from Chapter 3 in MRCS. Lastly, large language models (LLM) for high level synthesis [105] with ternary verilog in MRCS will be explored to speed up the design process.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

—”The physical world remains stubbornly analog .. they must ultimately rely on analog interfaces to digitize real-world information”

Prof. Murmann and Prof. Hoefflinger in [301]

5

Discussion

5.1 Towards a ternary technology stack

Binary computing has enormous inertia. It is a multi-billion ecosystem, with fabrication processes, software algorithms and education being optimized for it. Disrupting this ecosystem with radix-3 logic to overcome the three scaling walls is unlikely to happen in the next decade. Technology roadmaps are planned well in advance and new technologies are introduced in small numbers initially. The slow transition path might be similar to the MVL adoption paths of the memory and communication industry. Fortunately, binary is a subset of ternary so the transition path can be smooth. The same ternary logic design can be implemented with ternary signals or with BCT signals, as demonstrated with MRCS.

There are several steep challenges and milestones to beat for radix-3 logic to be competitive in practice. Perhaps the largest challenge is to make a power, performance, and area (PPA) competitive balanced ternary full adders. They are ubiquitous logic gates in CPU designs. This would require novel switching devices that can be fabricated with a CMOS-compatible process. Another steep challenge is to design production-ready ternary logic chips and demonstrate the theoretical merits at the system level. An example could be the low-power IoT industry with a 5-trit ternary microcontroller and compare it to an 8-bit equivalent. The goal for radix-3 is to be competitive on both cost and specs at mass-production scale for products that value efficiency more than simplicity. A major milestone to overcome is the binary RISC-V microcontrollers by WinChipHead ((WCH) for \$0.10 per chip [302]. Another milestone is the simplicity of designing microcontrollers with hardware description languages (HDL) like Silice which can describe a binary dual-core RISC-V CPU with just 120 lines of code [303]. Despite these steep challenges

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

The memory wall shows that there is a need for a better balance between communication and computation. Perhaps the greatest inspiration for modern IC's is Nature's balanced performance/watt solution: the brain [112]. In the long term a return to analog computing or a best-of-both-worlds [304] would make sense, with radix-3 being the first step towards a higher radix compute paradigm. A higher radix would benefit various emerging computing fields such as probabilistic, neuromorphic, optical and quantum computing. The mechanisms for computing in these fields are not tied to binary signals or encoding. A full embrace of the ternary compute paradigm requires reinventing/optimizing the technology stack and EDA flows presented in Fig. 1.2. This tremendous exercise touches all aspects of ternary computing: From programming language constructs and ternary-aware compilers, data structures, algorithms and operating systems to ISA, ternary VLSI and physical devices that work optimally with ternary signals.

5.2 Open questions

In 1977 Hamacher and Vranesic stated that ternary research needs to satisfy 3 factors to be a serious alternative to binary [153]: intellectual challenge, physical feasibility and applicability. This thesis showed it was worthwhile to view computing ideas and theories pioneered in the 1950's through a modern lens. There is a clear demand for more compute power in smaller, lower power form factors. There is nothing fundamentally blocking ternary computers and no computing limits are reached. New devices, circuits and systems are certainly waiting to be discovered. Five large open questions have been identified to continue this research:

1. What is needed to enrich hardware description languages (HDL) and high level synthesis (HLS) tools to describe and control the new capabilities radix-3 offers?
2. Does radix-3 reduce design complexity of the asynchronous compute paradigm as synchronization mechanisms such as the {read lock, no lock, write lock} can be captured in a single trit?
3. How can one or more layers of multi-state memristors, a memory controller and multi-threshold CNTFET logic be integrated in a 3D SoC workflow [305]?
4. What are the practical challenges and benefits of synchronous designs with a ternary clock, especially with respect to power consumption?
5. How does BCT compare to binary at the system level when considering a functionally equivalent CPU design on older process nodes using VLSI metrics?

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

“The key message for the next 50 years is ‘there is a lot of room at the bottom’. the message for the next 50 years is ‘there is a lot of room between devices’ or ‘there is a lot of slack in wires’”

P. Ruch et al., IBM Research [10]

—*“An early mentor told me to run towards problems, because that’s where you’ll find opportunity”*

Lisa Su, CEO of AMD

6

Conclusion

The aim of this PhD project was to revisit Atanasoff's and von Neumann's 80 year old conclusion and determine if radix-2 (binary) is still the better radix choice for computing given new device and material advancements. The full answer is presented over four chapters in this thesis and is summarized below.

The blunt conclusion is that radix-2 can no longer be considered the most efficient radix for computing. It remains the simplest radix and is therefore unlikely to become obsolete. The nuanced conclusion is that logic is still radix-2 and that memory and communication have shifted to MVL in the last 20 years for efficiency reasons. Post-binary examples include the latest SSD's, SD-cards, GDDR7 DRAM, embedded RRAM and communication standards like USB, I3C, Bluetooth EDR, WIFI and Ethernet. The fragmented situation of logic being binary and memory and communication being non-binary requires continuous and costly signal conversions. The heart of the problem is the continued bi-stable device-centric scaling which became inefficient after Dennard-scaling stopped in 2005. Under utilization of transistors (dark silicon) and the increasing energy and delay gap between computation on the ALU and communication with the ALU shows the need for more efficient computation using a higher radix rather than faster computation with radix-2. Fundamental and practical limits like Shannon's limit, Rent's rule, material and area limits at high frequencies further support a higher radix approach.

The radix economy theorem shows mathematically that radix-3 (ternary) is the closest discrete radix to the optimum e , but only when considering that the number of devices and the number of symbols per device are equally costly. This model has been shown to be accurate with memory devices which can hold multiple states (symbols) per device. For logic this cost model seems wrong as encoding circuit representations with CMOS is done in radix-1 with PMOS and NMOS devices each encoding a state. Ternary is the first

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

member of the MVL family and also the first odd radix. Shannon demonstrated in the 1950's that balanced arithmetic around zero with odd radixes is incredibly efficient and elegant for computation. Several benefits of ternary have been discussed in this thesis which were categorized around seven application domains (7 C's).

Central in this thesis is addressing the simulation, implementation and verification of radix-3 circuits. Measurements and experiments with multi-stable devices are still under reported in literature. The absence of higher radix EDA tools and modern MVL synthesis algorithms has consistently been mentioned as a knowledge gap. Another lacuna is fair radix comparison and system level benchmarking. Several recommendations were proposed in the thesis including feature parity and optimal pin utilization. The first tool developed was uMemristorToolbox, an open source software framework to experiment with multi-state memristive devices. Using the tool a non-volatile ternary memory controller was made and a multi-state RRAM development platform. Initial experiments confirm that radix-3 memory with memristors is both feasible and low-cost. More work is needed at larger scale with more device, chip and batch variance under wider operating conditions.

The second tool developed was Mixed Radix Circuit Synthesizer (MRCS), the first browser-based EDA tool to design and verify binary, ternary and hybrid (mixed radix) circuits. It features a novel MVL circuit synthesis algorithm with HSPICE and verilog output targeting CMOS and multi-threshold CNTFET. Ternary circuits are automatically converted to BCT when targeting verilog such that designers can focus on functionality. Designs can be converted to ASIC with Openlane's RTL-to-GDS toolchain, HSPICE Simulator or to FPGA with Vivado's RTL-To-Bitstream toolchain. Binary-to-ternary and inverse hardware converters made with MRCS are discussed in the thesis which enable seamless interfacing between the two radixes. The tool was used to design REBEL-2, a novel balanced ternary CPU with RISC-V-like ISA. The REBEL-2 ISA can resolve²⁷₃₈ RV32i instructions with only 9 opcodes and is great for educational purposes. Four MRCS designs have been submitted for tape-out using TinyTapeout, a new and affordable tape-out service. Initial verification of a TinyTapeout 2 sample show correct functionality.

This thesis started with the question if a richer computer alphabet has the same benefits as it does for humans. Patterson and Hennessy [2, p. 68] answered this splendidly:

"Computer designers have a common goal: to find a language that makes it easy to build the hardware and the compiler while maximizing performance and minimizing cost and energy."

The REBEL-2 ISA and industry developments that use ternary or BCT signals show that a richer computer alphabet empowers the computer language to scale today's power, memory and EDA walls. It is time to pivot away from device-centric scaling and engage in efficiency-centric scaling using a higher radix. Radix-3 is the optimal radix and would be the prime choice for this new compute paradigm.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [1] J. V. Atanasoff, ‘Advent of electronic digital computing,’ *Annals of the History of Computing*, vol. 6, no. 3, pp. 229–282, 1984. DOI: [10.1109/MAHC.1984.10028](https://doi.org/10.1109/MAHC.1984.10028).
- [2] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*. Morgan Kaufmann Publishers Inc., 2020, ISBN: 9780128203316.
- [3] M. Tomasello, *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, 2003, ISBN: 9780674010307. [Online]. Available: <http://www.jstor.org/stable/j.ctv26070v8>.
- [4] H. K. Pae, ‘The alphabet,’ in *Script Effects as the Hidden Drive of the Mind, Cognition, and Culture*. Springer, 2020. DOI: [10.1007/978-3-030-55152-0_4](https://doi.org/10.1007/978-3-030-55152-0_4).
- [5] J. Man, ‘Alpha beta,’ in *How 26 Letters Shaped The Western World*. John Wiley & Sons, 2001, ISBN: 978-0-471-41574-9.
- [6] D. Crystal, ‘The Cambridge encyclopedia of language,’ in Cambridge University Press, 2010, vol. 3, ISBN: 978-0-521-73650-3.
- [7] I. L. Markov, ‘Limits on fundamental limits to computation,’ *Nature*, vol. 512, no. 7513, pp. 147–154, Aug. 2014. DOI: [10.1038/nature13570](https://doi.org/10.1038/nature13570).
- [8] T. Mudge, ‘Power: A first-class architectural design constraint,’ *Computer*, vol. 34, no. 4, pp. 52–58, 2001. DOI: [10.1109/2.917539](https://doi.org/10.1109/2.917539).
- [9] S. Moore and D. Greenfield, ‘The next resource war: Computation vs. Communication,’ in *Proceedings International Workshop on System Level Interconnect Prediction*, Association for Computing Machinery, 2008, pp. 81–86. DOI: [10.1145/1353610.1353627](https://doi.org/10.1145/1353610.1353627).
- [10] P. Ruch, T. Brunschwiler, W. Escher, S. Paredes and B. Michel, ‘Toward five-dimensional scaling: How density improves efficiency in future computers,’ *IBM Journal of Research and Development*, vol. 55, no. 5, pp. 1–13, 2011. DOI: [10.1147/JRD.2011.2165677](https://doi.org/10.1147/JRD.2011.2165677).
- [11] G. Ifrah, *The universal history of computing: from the abacus to the quantum computer*. Wiley, 2001, ISBN: 978-0471441472.
- [12] W. Buchholz, ‘Fingers or fists? (the choice of decimal or binary representation),’ *Communications of the ACM*, vol. 2, no. 12, pp. 3–11, 1959. DOI: [10.1145/368518.368529](https://doi.org/10.1145/368518.368529).
- [13] N. H. McCoy, ‘Introduction to modern algebra,’ in Allyn and Bacon, 1968. DOI: [10.1145/368518.368529](https://doi.org/10.1145/368518.368529).
- [14] K. Rupp, *48 years of microprocessor trend data*, 2020. [Online]. Available: [10.5281/zenodo.3947823](https://doi.org/10.5281/zenodo.3947823).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [15] S. K. Moore and D. Schneider, *The state of the transistor in 3 charts*, 2022. [Online]. Available: <https://spectrum.ieee.org/transistor-density>.
- [16] K. Flamm, *Measuring Moore's law: Evidence from price, cost, and quality indexes*, 2018. [Online]. Available: [10.3386/w24553](https://doi.org/10.3386/w24553).
- [17] G. E. Moore, 'Cramming more components onto integrated circuits,' *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, 2006. DOI: [10.1109/NSSC.2006.4785860](https://doi.org/10.1109/NSSC.2006.4785860).
- [18] R. Dennard, F. Gaenslen, H.-N. Yu, V. Rideout, E. Bassous and A. LeBlanc, 'Design of ion-implanted MOSFET's with very small physical dimensions,' *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974. DOI: [10.1109/JSSC.1974.1050511](https://doi.org/10.1109/JSSC.1974.1050511).
- [19] T. H. Bullock, M. V. L. Bennett, D. Johnston, R. Josephson, E. Marder and R. D. Fields, 'The neuron doctrine, redux,' *Science*, vol. 310, no. 5749, pp. 791–793, 2005. DOI: [10.1126/science.1114394](https://doi.org/10.1126/science.1114394).
- [20] A. Mehonic and A. J. Kenyon, 'Brain-inspired computing needs a master plan,' *Nature*, vol. 604, no. 7905, pp. 255–260, Apr. 2022. DOI: [10.1038/s41586-021-04362-w](https://doi.org/10.1038/s41586-021-04362-w).
- [21] L. Baudry, I. Lukyanchuk and V. M. Vinokur, 'Ferroelectric symmetry-protected multibit memory cell,' *Scientific Reports*, vol. 7, no. 1, p. 42196, Feb. 2017. DOI: [10.1038/srep42196](https://doi.org/10.1038/srep42196).
- [22] B. Sengupta and M. B. Stemmler, 'Power consumption during neuronal computation,' *Proceedings of the IEEE*, vol. 102, no. 5, pp. 738–750, 2014. DOI: [10.1109/JPROC.2014.2307755](https://doi.org/10.1109/JPROC.2014.2307755).
- [23] C. E. Shannon, 'A symmetrical notation for numbers,' *The American Mathematical Monthly*, vol. 57, no. 2, pp. 90–93, 1950. DOI: [10.1080/00029890.1950.11999490](https://doi.org/10.1080/00029890.1950.11999490).
- [24] D. E. Knuth, 'Positional number systems,' in *The Art of Computer Programming: Vol. 2 Seminumerical Algorithms*, Addison-Wesley, 1969, ch. 4.1, pp. 207–209, ISBN: 0-201-03802-1.
- [25] R. P. Feynman, *Feynman Lectures on Computation (Frontiers in Physics)*, T. Hey and A. R. W., Eds. CRC Press, 2000, vol. 1, ISBN: 978-0-738-20296-9.
- [26] N. P. Brousentsov, S. P. Maslov, J. Ramil Alvarez and E. A. Zhogolev, 'Development of ternary computers at Moscow State University,' *Russian Virtual Computer Museum*, 2002. [Online]. Available: <https://www.computer-museum.ru/english/setun.htm>.
- [27] F. Hunger, *Setun: An inquiry into the Soviet Ternary Computer*. Institut für Buchkunst Leipzig, 2007.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [28] J. R. Brusentsov N. P.; Alvarez, ‘Ternary computers: The Setun and the Setun 70,’ *Perspectives on Soviet and Russian Computing. SoRuCom 2006. IFIP Advances in Information and Communication Technology*, vol. 357, pp. 74–80, 2011. DOI: [10.1007/978-3-642-22816-2_10](https://doi.org/10.1007/978-3-642-22816-2_10).
- [29] G. Hills, C. Lau, A. Wright *et al.*, ‘Modern microprocessor built from complementary carbon nanotube transistors,’ *Nature*, vol. 572, no. 7771, pp. 595–602, Aug. 2019. DOI: [10.1038/s41586-019-1493-8](https://doi.org/10.1038/s41586-019-1493-8).
- [30] A. Raychowdhury and K. Roy, ‘Carbon-nanotube-based voltage-mode multiple-valued logic design,’ *IEEE Transactions on Nanotechnology*, vol. 4, no. 2, pp. 168–179, 2005. DOI: [10.1109/TNANO.2004.842068](https://doi.org/10.1109/TNANO.2004.842068).
- [31] M. D. Bishop, G. Hills, T. Srimani *et al.*, ‘Fabrication of carbon nanotube field-effect transistors in commercial silicon manufacturing facilities,’ *Nature Electronics*, vol. 3, no. 8, pp. 492–501, Aug. 2020. DOI: [10.1038/s41928-020-0419-7](https://doi.org/10.1038/s41928-020-0419-7).
- [32] K. E. Aasmundtveit, A. Roy and B. Q. Ta, ‘Direct integration of carbon nanotubes in CMOS, towards an industrially feasible process: A review,’ *IEEE Transactions on Nanotechnology*, vol. 19, pp. 113–122, 2020. DOI: [10.1109/TNANO.2019.2961415](https://doi.org/10.1109/TNANO.2019.2961415).
- [33] K. E. Aasmundtveit, B. Q. Ta, L. Lin, E. Halvorsen and N. Hoivik, ‘Direct integration of carbon nanotubes in Si microstructures,’ *Journal of Micromechanics and Microengineering*, vol. 22, no. 7, Jun. 2012. DOI: [10.1088/0960-1317/22/7/074006](https://doi.org/10.1088/0960-1317/22/7/074006).
- [34] L. Chua, ‘Memristor: The missing circuit element,’ *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971. DOI: [10.1109/TCT.1971.1083337](https://doi.org/10.1109/TCT.1971.1083337).
- [35] D. B. Strukov, G. S. Snider, D. R. Stewart and R. S. Williams, ‘The missing memristor found,’ *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008. DOI: [10.1038/nature06932](https://doi.org/10.1038/nature06932).
- [36] M. Rao, H. Tang, J. Wu *et al.*, ‘Thousands of conductance levels in memristors integrated on CMOS,’ *Nature*, vol. 615, no. 7954, pp. 823–829, Mar. 2023. DOI: [10.1038/s41586-023-05759-5](https://doi.org/10.1038/s41586-023-05759-5).
- [37] B. E. Carpenter, ‘Turing and ACE: Lessons from a 1946 computer design,’ *1992 CERN School of Computing*, 1991. DOI: [10.5170/CERN-1993-003.230](https://doi.org/10.5170/CERN-1993-003.230).
- [38] W. Phillips, J. V. Atanasoff, D. Michie, J. W. Mauchly, H. H. Goldstine and A. Goldstine, *The Advent of Electronic Computers*, B. Randell, Ed. Springer, 1973, pp. 287–347. DOI: [10.1007/978-3-642-96145-8_7](https://doi.org/10.1007/978-3-642-96145-8_7).
- [39] W. Aspray and A. Glaser, ‘History of binary and other nondecimal numeration,’ 1981, ISBN: 978-0938228004.
- [40] H. Iwai, ‘History of transistor invention: 75th anniversary,’ in *2022 IEEE 16th International Conference on Solid-State & Integrated Circuit Technology (ICSICT)*, 2022, pp. 1–10. DOI: [10.1109/ICSICT55466.2022.9963262](https://doi.org/10.1109/ICSICT55466.2022.9963262).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [41] P. Ball, ‘Polynesian people used binary numbers 600 years ago,’ *Nature*, Dec. 2013. DOI: [10.1038/nature.2013.14380](https://doi.org/10.1038/nature.2013.14380).
- [42] J. Ares, J. Lara, D. Lizcano and M. A. Martínez, ‘Who discovered the binary system and arithmetic? Did Leibniz plagiarize Caramuel?’ *Science and Engineering Ethics*, vol. 24, no. 1, pp. 173–188, Feb. 2018. DOI: [10.1007/s11948-017-9890-6](https://doi.org/10.1007/s11948-017-9890-6).
- [43] J. W. Shirley, ‘Binary Numeration before Leibniz,’ *American Journal of Physics*, vol. 19, no. 8, pp. 452–454, Nov. 1951. DOI: [10.1119/1.1933042](https://doi.org/10.1119/1.1933042).
- [44] A. M. Turing, ‘On computable numbers, with an application to the entscheidungsproblem,’ *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937. DOI: [10.1112/plms/s2-42.1.230](https://doi.org/10.1112/plms/s2-42.1.230).
- [45] B. Randell, ‘On Alan Turing and the origins of digital computers,’ 1972, pp. 1–20. [Online]. Available: <https://www.computerhistory.org/collections/catalog/102724643>.
- [46] C. E. Shannon, ‘A symbolic analysis of relay and switching circuits,’ *Transactions of the American Institute of Electrical Engineers*, vol. 57, no. 12, pp. 713–723, 1938. DOI: [10.1109/T-AIEE.1938.5057767](https://doi.org/10.1109/T-AIEE.1938.5057767).
- [47] J. Gilbey, ‘Biography: The ABC of computing,’ *Nature*, vol. 468, no. 7325, pp. 760–761, Dec. 2010. DOI: [10.1038/468760a](https://doi.org/10.1038/468760a).
- [48] V. Getov, ‘Insights into the origins of the IEEE computer society and the invention of electronic digital computing,’ *Computer*, vol. 54, pp. 13–18, Aug. 2021. DOI: [10.1109/MC.2021.3084067](https://doi.org/10.1109/MC.2021.3084067).
- [49] J. Smile, *The man who invented the computer: the biography of John Atanasoff, digital pioneer*. Doubleday, 2010, ISBN: 0385527136.
- [50] I. H. Anellis, ‘John Vincent Atanasoff: His place in the history of computer logic and technology,’ *Modern Logic*, vol. 7, no. 1, pp. 1–24, 1997. [Online]. Available: <https://projecteuclid.org/journals/modern-logic/volume-7/issue-1/John-Vincent-Atanasoff---his-place-in-the-history/rml/1204900339.full/>.
- [51] J. von Neumann, ‘First draft of a report on the EDVAC,’ *IEEE Annals of the History of Computing*, vol. 15, no. 4, pp. 27–75, 1993. DOI: [10.1109/85.238389](https://doi.org/10.1109/85.238389).
- [52] A. W. Burks, H. H. Goldstine and J. von Neumann, ‘Preliminary discussion of the logical design of an electronic computing instrument,’ in *Papers of John von Neumann on Computing and Computing Theory*, W. Aspray and A. W. Burks, Eds., MIT Press, 1987, pp. 97–142. DOI: <https://doi.org/10.5555/98326.98337>.
- [53] W. S. McCulloch and W. Pitts, ‘A logical calculus of the ideas immanent in nervous activity,’ *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943. DOI: [10.1007/BF02478259](https://doi.org/10.1007/BF02478259).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [54] F. P. Brooks, G. A. Blaauw and W. Buchholz, 'Processing data in bits and pieces,' *IRE Transactions on Electronic Computers*, vol. EC-8, no. 2, pp. 118–124, 1959. DOI: 10.1109/TEC.1959.5219512.
- [55] L. Shustek, 'An interview with Fred Brooks,' *Communications of the ACM*, vol. 58, no. 11, pp. 36–40, Oct. 2015, ISSN: 0001-0782. DOI: 10.1145/2822519.
- [56] R. Richards, *Arithmetic operations in digital computers*. D. van Nostrand Company Inc., 1955. [Online]. Available: https://archive.org/details/arithmetic_operations_in_digital_computers.
- [57] Engineering Research Associates Staff, *High-Speed Computing Devices*, C. Tompkins, J. Wakelin and W. Stifler Jr., Eds. McGraw-Hill, 1950, ISBN: 9780262050289.
- [58] W. H. Ware, 'Soviet computer technology—1959,' *Communications of the ACM*, 1960. DOI: 10.1145/367149.1047530.
- [59] Gartner, *Gartner forecasts worldwide semiconductor revenue to grow 13.6% in 2022*, 2022. [Online]. Available: <https://www.gartner.com/en/newsroom/press-releases/2022-04-26-gartner-forecasts-worldwide-semiconductor-revenue-to-grow-13-6-percent-in-2022>.
- [60] V. Smil, *Moore's curse. There is a dark side to the revolution in electronics: Unjustified technological expectations*, 2015. [Online]. Available: <https://spectrum.ieee.org/moores-curse>.
- [61] More Moore team, *International Roadmap for Devices and Systems: 2022 update More Moore*, 2022. [Online]. Available: https://irds.ieee.org/images/files/pdf/2022/2022IRDS_MM.pdf.
- [62] Beyond CMOS team, *International Roadmap for Devices and Systems: 2022 update Beyond CMOS and emerging materials integration*, 2022. [Online]. Available: https://irds.ieee.org/images/files/pdf/2022/2022IRDS_BC.pdf.
- [63] A. Kanduri, A. M. Rahmani, P. Liljeberg, A. Hemani, A. Jantsch and H. Tenhunen, *A Perspective on Dark Silicon*, A. M. Rahmani, P. Liljeberg, A. Hemani, A. Jantsch and H. Tenhunen, Eds. Cham: Springer, 2017, pp. 3–20. DOI: 10.1007/978-3-319-31596-6_1.
- [64] P. Bose, 'Power wall,' in *Encyclopedia of Parallel Computing*, D. Padua, Ed. Boston, MA: Springer, 2011, pp. 1593–1608. DOI: 10.1007/978-0-387-09766-4_499.
- [65] M. Bohr, 'A 30 year retrospective on Dennard's MOSFET scaling paper,' *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 1, pp. 11–13, 2007. DOI: 10.1109/N-SSC.2007.4785534.
- [66] R. Ho, K. Mai and M. Horowitz, 'The future of wires,' *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490–504, 2001. DOI: 10.1109/5.920580.
- [67] S. M. Sze, *Setun, An inquiry into the Soviet Ternary Computer*. Wiley and Sons,

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [68] A. Agarwal, P. C. Pradhan and B. P. Swain, ‘From FET to SET: A review,’ in *Advances in Electronics, Communication and Computing*, A. Kalam, S. Das and K. Sharma, Eds., Springer, 2018, pp. 199–209, ISBN: 978-981-10-4765-7.
- [69] L. Qin, C. Li, Y. Wei *et al.*, ‘Recent developments in negative capacitance gate-all-around field effect transistors: A review,’ *IEEE Access*, vol. 11, pp. 14 028–14 042, 2023. DOI: 10.1109/ACCESS.2023.3243697.
- [70] W. R. Deal, K. Leong, W. Yoshida, A. Zamora and X. B. Mei, ‘InP HEMT integrated circuits operating above 1,000 GHz,’ in *IEEE International Electron Devices Meeting (IEDM)*, 2016, pp. 29.1.1–29.1.4. DOI: 10.1109/IEDM.2016.7838502.
- [71] D. C. Sekar, A. Naeemi, R. Sarvari, J. A. Davis and J. D. Meindl, ‘Intsim: A cad tool for optimization of multilevel interconnect networks,’ in *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, 2007, pp. 560–567. DOI: 10.1109/ICCAD.2007.4397324.
- [72] Z. Tokei, *Scaling the back end of line – a toolbox filled with new processes, boosters and conductors*, 2019. [Online]. Available: <https://www.imec-int.com%7C/en/imec-magazine/imec-magazine-september-2019/scaling-the-beol-a-toolbox-filled-with-new-processes-boosters-and-conductors>.
- [73] D. Brooks, P. Bose, S. Schuster *et al.*, ‘Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors,’ *IEEE Micro*, vol. 20, no. 6, pp. 26–44, 2000. DOI: 10.1109/40.888701.
- [74] A. Ranjan, *Micro-architectural exploration for low power design*, 2015. [Online]. Available: <https://semiengineering.com/micro-architectural-exploration-for-low-power-design/>.
- [75] W. Nakayama, ‘Heat in computers: Applied heat transfer in information technology,’ *Journal of Heat Transfer*, vol. 136, Jan. 2014. DOI: 10.1115/1.4025377.
- [76] G. M. Gilson, S. J. Pickering, D. B. Hann and C. Gerada, ‘Piezoelectric fan cooling: A novel high reliability electric machine thermal management solution,’ *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 4841–4851, 2013. DOI: 10.1109/TIE.2012.2224081.
- [77] W. Wulf and S. McKee, ‘Hitting the memory wall: Implications of the obvious,’ *Computer Architecture News*, vol. 23, Jan. 1996.
- [78] J. L. Hennessy and D. A. Patterson, *Computer Architecture, A Quantitative Approach*. Morgan Kaufmann, 2019, ISBN: 9780128119051.
- [79] M. Jung, S. A. McKee, C. Sudarshan, C. Dropmann, C. Weis and N. Wehn, ‘Driving into the Memory Wall: The role of memory for advanced driver assistance systems and autonomous driving,’ in *Proceedings of the International Symposium*

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

-
- [80] X. Zou, S. Xu, X. Chen, L. Yan and Y. Han, ‘Breaking the von Neumann bottleneck: Architecture-level processing-in-memory technology,’ *Science China Information Sciences*, vol. 64, no. 6, p. 160404, Apr. 2021. DOI: 10.1007/s11432-020-3227-1.
 - [81] A. Spessot and H. Oh, ‘1T-1C Dynamic Random Access Memory status, challenges, and prospects,’ *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1382–1393, 2020. DOI: 10.1109/TED.2020.2963911.
 - [82] T. Hiramoto, ‘Five nanometre CMOS technology,’ *Nature Electronics*, vol. 2, no. 12, pp. 557–558, Dec. 2019. DOI: 10.1038/s41928-019-0343-x.
 - [83] J. Ryckaert, P. Weckx and S. M. Salahuddin, ‘3 - SRAM technology status and perspectives,’ in *Semiconductor Memories and Systems*, A. Redaelli and F. Pellizzer, Eds., Woodhead Publishing, 2022, pp. 55–86. DOI: 10.1016/B978-0-12-820758-1.00010-8.
 - [84] S. Rusu, H. Muljono and B. Cherkauer, ‘Itanium 2 processor 6M: Higher frequency and larger L3 cache,’ *Micro, IEEE*, vol. 24, pp. 10–18, Apr. 2004. DOI: 10.1109/MM.2004.1289279.
 - [85] H. M. D. Kabir and M. Chan, ‘SRAM precharge system for reducing write power,’ *HKIE Transactions*, vol. 22, no. 1, pp. 1–8, 2015. DOI: 10.1080/1023697X.2014.970761.
 - [86] Intel, *Intel optane persistent memory*, 2020. [Online]. Available: <https://www.intel.com/content/www/us/en/developer/articles/technical/aerospike-pmdk-intel-optane-persistent-memory.html>.
 - [87] W. Wan, R. Kubendran, C. Schaefer *et al.*, ‘A compute-in-memory chip based on Resistive Random-Access Memory,’ *Nature*, vol. 608, no. 7923, pp. 504–512, Aug. 2022. DOI: 10.1038/s41586-022-04992-8.
 - [88] M. Lapedus, *Big trouble at 3nm*, 2018. [Online]. Available: <https://semiengineering.com/big-trouble-at-3nm>.
 - [89] O. Burkacky, M. de Jong and J. Dragon, *Strategies to lead in the semiconductor world*, 2022. [Online]. Available: <https://semiengineering.com/micro-architecture-exploration-for-low-power-design/>.
 - [90] T. Ajayi, D. Blaauw, T.-B. Chan *et al.*, ‘Openroad: Toward a self-driving, open-source digital layout implementation tool chain,’ in *Proc. Government Microcircuit Applications and Critical Technology Conference*, 2019, pp. 1105–1110.
 - [91] O. Andreas, *Intelligent design of electronic assets (IDEA) & posh open source hardware (POSH)*, 2017. [Online]. Available: <https://www.darpa.mil/attachments/>

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Ternary Computer, 1th, F. Maloberti and A. C. Davies, Eds. River, 2016, ISBN: 978-8793379718.

Bibliography

- [93] D. Abercrombie and M. White, *IC design: Preparing for the next node*, 2019. [Online]. Available: <https://resources.sw.siemens.com/en-US/white-paper-ic-design-preparing-for-the-next-node#disw-fulfillment-form>.
- [94] E. Sperling, *Design rule complexity rising*, 2018. [Online]. Available: <https://semiengineering.com/design-rule-complexity-rising/>.
- [95] F. Faggin, ‘The making of the first microprocessor,’ *IEEE Solid-State Circuits Magazine*, vol. 1, no. 1, pp. 8–21, 2009. DOI: 10.1109/MSSC.2008.930938.
- [96] A. Sangiovanni-Vincentelli, ‘The tides of EDA,’ *IEEE Design & Test of Computers*, vol. 20, no. 6, pp. 59–75, 2003. DOI: 10.1109/MDT.2003.1246165.
- [97] E. Dubrova, ‘Multiple-Valued Logic in VLSI: Challenges and opportunities,’ *Proceedings of NORCHIP’99*, Nov. 1999.
- [98] T. Hossain, S. M. M. Ahsan and T. Hoque, ‘Potential and pitfalls of Multi-Valued Logic circuits for hardware security,’ in *Dallas Circuits and Systems Conference (DCAS)*, 2023, pp. 1–6. DOI: 10.1109/DCAS57389.2023.10130261.
- [99] E. Dubrova, ‘Multiple-Valued Logic synthesis and optimization,’ in *Logic Synthesis and Verification*, S. Hassoun and T. Sasao, Eds. Boston, MA: Springer, 2002, pp. 89–114. DOI: 10.1007/978-1-4615-0817-5_4.
- [100] T. Ajayi, V. A. Chhabria, M. Fogaça *et al.*, ‘Toward an open-source digital flow: First learnings from the OpenROAD project,’ in *Proceedings Annual Design Automation Conference 2019*, New York, NY, USA: Association for Computing Machinery, 2019. DOI: 10.1145/3316781.3326334.
- [101] M. Shalan and T. Edwards, ‘Building OpenLANE: A 130nm OpenROAD-based tapeout- proven flow,’ in *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2020, pp. 1–6, ISBN: 978-1-6654-2324-3.
- [102] A. Olofsson, W. Ransohoff and N. Moroze, ‘A distributed approach to silicon compilation,’ in *Proceedings ACM/IEEE Design Automation Conference*, New York, NY, USA: Association for Computing Machinery, 2022, pp. 1343–1346. DOI: 10.1145/3489517.3530673.
- [103] ‘ASAP7: A 7-nm finFET predictive process design kit,’ *Microelectronics Journal*, vol. 53, pp. 105–115, 2016. DOI: 10.1016/j.mejo.2016.04.006.
- [104] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi and J. Kepner, ‘AI and ML accelerator survey and trends,’ in *2022 IEEE High Performance Extreme*

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [105] J. Blocklove, S. Garg, R. Karri and H. Pearce, 'Chip-chat: Challenges and opportunities in conversational hardware design,' in *Workshop on Machine Learning for CAD (MLCAD)*, 2023, pp. 1–6. doi: 10.1109/MLCAD58807.2023.10299874.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [207] B. Cambou and M. Orlowski, ‘PUF designed with Resistive RAM and ternary states,’ in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, Association for Computing Machinery, 2016. doi: 10.1145/2897795.2897808.
- [208] P. G. Flikkema, J. Palmer, T. Yalcin and B. Cambou, ‘Dynamic computational diversity with multi-radix logic and memory,’ in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–6. doi: 10.1109/HPEC43674.2020.9286255.
- [209] S. Assiri, B. Cambou, D. D. Booher, D. Ghanai Miandoab and M. Mohammadinodoushan, ‘Key exchange using ternary system to enhance security,’ in *Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0488–0492. doi: 10.1109/CCWC.2019.8666511.
- [210] B. Parhami, ‘Truncated ternary multipliers,’ *IET Computers & Digital Techniques*, vol. 9, no. 2, pp. 101–105, 2015. doi: 10.1049/iet-cdt.2013.0133.
- [211] S. Zhu, L. H. K. Duong, H. Chen, D. Liu and W. Liu, ‘FAT: An in-memory accelerator with fast addition for ternary weight Neural Networks,’ *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 781–794, Mar. 2023. doi: 10.1109/TCAD.2022.3184276.
- [212] P. Schuddinck, F. M. Bufler, Y. Xiang *et al.*, ‘Ppac of sheet-based cfet configurations for 4 track design with 16nm metal pitch,’ in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, 2022, pp. 365–366. doi: 10.1109/VLSITechnologyandCir46769.2022.9830492.
- [213] Z. G. Vranesic and K. C. Smith, ‘Engineering aspects of multi-valued logic systems,’ *Computer*, vol. 7, no. 9, pp. 34–41, 1974. doi: 10.1109/MC.1974.6323306.
- [214] B. Behin-Aein, D. Datta, S. Salahuddin and S. Datta, ‘Proposal for an all-spin logic device with built-in memory,’ *Nature Nanotechnology*, vol. 5, no. 4, pp. 266–270, Apr. 2010. doi: 10.1038/nnano.2010.31.
- [215] D. Etiemble, ‘Common fallacies about multivalued circuits,’ *Asian Journal of Research in Computer Science*, vol. 12, no. 4, pp. 67–83, Dec. 2021. doi: 10.9734/ajrcos/2021/v12i430295.
- [216] D. Etiemble, *Technologies and computing paradigms: Beyond moore’s law?* 2022. arXiv: 2206.03201 [cs.ET].
- [217] S. Shin, E. Jang, J. W. Jeong and K. R. Kim, ‘CMOS-compatible ternary device platform for physical synthesis of Multi-Valued Logic circuits,’ in *2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL)*, 2017, pp. 284–289. doi: 10.1109/ISMVL.2017.48.
- [218] Skywater Technologies, *Carbon Nanotube SoCs: High-density, stackable SoCs with Carbon Nanotube CMOS FETs + ReRAM*, 2020. [Online]. Available: <https://www.skywatertechnology.com/carbon-nanotubes/>.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [219] M. Forghani and B. Razavi, ‘Circuit bandwidth requirements for NRZ and PAM4 signals,’ in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 990–994. DOI: [10.1109/ISCAS48785.2022.9937588](https://doi.org/10.1109/ISCAS48785.2022.9937588).
- [220] B. D. Madhuri and S. Sunithamani, ‘Crosstalk noise analysis of on-chip interconnects for ternary logic applications using FDTD,’ *Microelectronics Journal*, vol. 93, 2019. DOI: [10.1016/j.mejo.2019.104633](https://doi.org/10.1016/j.mejo.2019.104633).
- [221] S. Pathania, S. Kumar and R. Sharma, ‘Crosstalk analysis for rough copper interconnects considering ternary logic,’ in *IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, 2018, pp. 1–3. DOI: [10.1109/EDAPS.2018.8680906](https://doi.org/10.1109/EDAPS.2018.8680906).
- [222] M. Takbiri, K. Navi and R. F. Mirzaee, ‘Noise margin calculation in Multiple-Valued Logic,’ in *International Conference on Computer and Knowledge Engineering (ICCKE)*, 2020, pp. 250–255. DOI: [10.1109/ICCKE50421.2020.9303638](https://doi.org/10.1109/ICCKE50421.2020.9303638).
- [223] D. Etiemble, ‘Why M-valued circuits are restricted to a small niche,’ *J. Multiple Valued Log. Soft Comput.*, vol. 9, no. 1, pp. 109–123, 2003. [Online]. Available: <https://www.oldcitypublishing.com/journals/mvlsc-home/mvlsc-issue-contents/mvlsc-volume-9-number-1-2003/mvlsc-9-1-p-109-123/>.
- [224] D. Etiemble and R. A. Jaber, ‘Design of (3,2) and (4,2) CNTFET ternary counters for multipliers,’ vol. 16, pp. 103–118, Jul. 2023. DOI: [10.9734/ajrcos/2023/v16i3349](https://doi.org/10.9734/ajrcos/2023/v16i3349).
- [225] D. Etiemble, ‘Post algebras and ternary adders,’ *Journal of Electrical Systems and Information Technology*, vol. 10, no. 1, p. 20, Mar. 2023. DOI: [10.1186/s43067-023-00088-z](https://doi.org/10.1186/s43067-023-00088-z).
- [226] D. Etiemble, *Multivalued circuits and interconnect issues*, 2020. arXiv: 2012.01267 [cs.AR].
- [227] D. Etiemble, ‘On the performance of Multivalued Integrated Circuits: Past, present and future,’ in *Proceedings International Symposium on Multiple-Valued Logic*, 1992, pp. 156–164. DOI: [10.1109/ISMVL.1992.186790](https://doi.org/10.1109/ISMVL.1992.186790).
- [228] H. Sutter, *The free lunch is over; a fundamental turn toward concurrency in software*, 2005. [Online]. Available: <http://www.gotw.ca/publications/concurrency-ddj.htm>.
- [229] L. O. Chua, *The chua lectures - part 1 : Once over lightly*, 2015. [Online]. Available: <https://youtu.be/B9Z2Ktacd4s&t=665>.
- [230] M.-K. Song, J.-H. Kang, X. Zhang *et al.*, ‘Recent advances and future prospects for memristive materials, devices, and systems,’ *ACS Nano*, vol. 17, no. 13, pp. 11994–12039, Jul. 2023, ISSN: 1936-0851. DOI: [10.1021/acsnano.3c03505](https://doi.org/10.1021/acsnano.3c03505).
- [231] M. Lanza, A. Sebastian, W. D. Lu *et al.*, ‘Memristive technologies for data storage, computation, encryption, and radio-frequency communication,’ *Science*, vol. 376, no. 6597, eabj9979, 2022. DOI: [10.1126/science.abbj9979](https://doi.org/10.1126/science.abbj9979).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [232] S. Technologies, *Skywater open source pdk*, 2020. [Online]. Available: <https://github.com/google/skywater-pdk>.
- [233] R. S. Williams, ‘How we found the missing memristor,’ *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, 2008. DOI: [10.1109/MSPEC.2008.4687366](https://doi.org/10.1109/MSPEC.2008.4687366).
- [234] W. Shim, J. Meng, X. Peng, J. Seo and S. Yu, ‘Impact of multilevel retention characteristics on RRAM based DNN inference engine,’ in *IEEE International Reliability Physics Symposium, IRPS 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021. DOI: [10.1109/IRPS46558.2021.9405210](https://doi.org/10.1109/IRPS46558.2021.9405210).
- [235] Fujitsu Semiconductor Memory Solution Limited, *ReRAM overview (white paper)*, 2023. [Online]. Available: https://www.fujitsu.com/jp/group/fsm/en/products/reram/ReRAM_whitepaper_2023e.pdf.
- [236] Fujitsu Semiconductor Memory Solution Limited, *MIKROE-3641 ReRAM click development board*, 2023. [Online]. Available: <https://www.mikroe.com/reram-click>.
- [237] Knowm Inc., *Knowm Shop*, 2019. [Online]. Available: <https://knowm.com/collections/all>.
- [238] Knowm Inc., *Knowm SDC datasheet*, 2019. [Online]. Available: https://knowm.org/downloads/Knowm_Memristors.pdf.
- [239] S. K. Kingra, V. Parmar, D. Verma *et al.*, ‘Fully binarized, parallel, RRAM-based computing primitive for in-memory similarity search,’ *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 1, pp. 46–50, 2023. DOI: [10.1109/TCSII.2022.3207378](https://doi.org/10.1109/TCSII.2022.3207378).
- [240] J. B. Nilsen, ‘Memristor Implementation of a Ternary Storage Circuit,’ M.S. thesis, USN, Norway, 2020.
- [241] M. S. Virk, ‘Memristor Development Platform - Dual Source Control For Implementations of Multistate Memristive Memory,’ M.S. thesis, USN, Norway, 2022.
- [242] M. D. Pickett, D. B. Strukov, J. L. Borghetti *et al.*, ‘Switching dynamics in titanium dioxide memristive devices,’ *Journal of Applied Physics*, vol. 106, no. 7, Oct. 2009. DOI: [10.1063/1.3236506](https://doi.org/10.1063/1.3236506).
- [243] K. A. Campbell, ‘Self-Directed Channel memristor for high temperature operation,’ *Microelectronics journal*, vol. 59, pp. 10–14, 2017. DOI: [10.1016/j.mejo.2016.11.006](https://doi.org/10.1016/j.mejo.2016.11.006).
- [244] S. Stathopoulos, A. Khiat, M. Trapatseli *et al.*, ‘Multibit memory operation of metal-oxide bi-layer memristors,’ *Nature Scientific Reports*, vol. 7, p. 17532, 2017. DOI: [10.1038/s41598-017-17785-1](https://doi.org/10.1038/s41598-017-17785-1).
- [245] J. Geler-Kremer, F. Eltes, P. Stark *et al.*, ‘A ferroelectric multilevel non-volatile photonic phase shifter,’ *Nature Photonics*, vol. 16, no. 7, pp. 491–497, Jul. 2022. DOI: [10.1038/s41566-022-01003-0](https://doi.org/10.1038/s41566-022-01003-0).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [246] N. TaheriNejad and D. Radakovits, ‘From behavioral design of memristive circuits and systems to physical implementations,’ *IEEE Circuits and Systems Magazine*, vol. 19, no. 4, pp. 6–18, 2019. doi: 10.1109/MCAS.2019.2945209.
- [247] Knowm Inc., *Memristor discovery github project*, 2019. [Online]. Available: <https://github.com/knowm/memristor-discovery> (visited on 01/03/2020).
- [248] S. Bos, *uMemristorToolbox github repository*, 2022. [Online]. Available: <https://github.com/aiunderstand/uMemristorToolbox>.
- [249] A. Nugent, *Knowm memristor discovery manual*, 2019.
- [250] A. Jagath, C. Leong, N. Thulasiraman and H. Almurib, ‘An insight into physics based RRAM models – a review,’ *The Journal of Engineering*, vol. 2019, May 2019. doi: 10.1049/joe.2018.5234.
- [251] S. Kvatinsky, E. G. Friedman, A. Kolodny and U. C. Weiser, ‘Team: Threshold adaptive memristor model,’ *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 211–221, 2013. doi: 10.1109/TCSI.2012.2215714.
- [252] S. Kvatinsky, M. Ramadan, E. G. Friedman and A. Kolodny, ‘VTEAM: A general model for voltage-controlled memristors,’ *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015. doi: 10.1109/TCSII.2015.2433536.
- [253] D. Biolek, Z. Kolka, V. Biolková, Z. Biolek and S. Kvatinsky, ‘(V)TEAM for SPICE simulation of memristive devices with improved numerical performance,’ *IEEE Access*, vol. 9, pp. 30 242–30 255, 2021. doi: 10.1109/ACCESS.2021.3059241.
- [254] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny and U. C. Weiser, ‘Memristor-Based Material Implication (IMPLY) logic: Design principles and methodologies,’ *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2054–2066, 2014. doi: 10.1109/TVLSI.2013.2282132.
- [255] S. Kvatinsky, D. Belousov, S. Liman *et al.*, ‘MAGIC—Memristor-Aided Logic,’ *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014. doi: 10.1109/TCSII.2014.2357292.
- [256] T. Molter, *Knowm SDC datasheet*, 2017. [Online]. Available: <https://knowm.org/memristor-models-in-ltspice/>.
- [257] D. Radakovits and N. TaheriNejad, ‘Implementation and characterization of a memristive memory system,’ in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 2019, pp. 1–4. doi: 10.1109/CCECE.2019.8861788.
- [258] G. Tobey, J. Graeme and L. Huelsman, *Burr-Brown Operational Amplifiers Design and Applications*. McGraw Hill, 1971, ISBN: 978-0070649170.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [259] C. E. Shannon, ‘The synthesis of two-terminal switching circuits,’ *The Bell System Technical Journal*, vol. 28, no. 1, pp. 59–98, 1949. DOI: [10.1002/j.1538-7305.1949.tb03624.x](https://doi.org/10.1002/j.1538-7305.1949.tb03624.x).
- [260] E. Fegri, ‘Design of a Balanced Ternary Tridirectional Loadable Counter Using CNTFETs,’ M.S. thesis, USN, Norway, 2022.
- [261] S. Bos, *Mixed Radix Circuit Synthesizer (MRCS) github repository*, 2022. [Online]. Available: <https://github.com/aiunderstand/MixedRadixCircuitSynthesis>.
- [262] S. Bos, *Mixed Radix Circuit Synthesis website*, 2022. [Online]. Available: <https://ternaryresearch.com/webgl/mixedradixcircuitsynthesizer/>.
- [263] R. K. Brayton, ‘The future of logic synthesis and verification,’ in *Logic Synthesis and Verification*. Boston, MA: Springer, 2002, pp. 403–434. DOI: [10.1007/978-1-4615-0817-5_15](https://doi.org/10.1007/978-1-4615-0817-5_15).
- [264] J. Deng and H.-S. P. Wong, ‘A compact SPICE model for Carbon-Nanotube Field-Effect Transistors including nonidealities and its application—part i: Model of the intrinsic channel region,’ *IEEE Transactions on Electron Devices*, vol. 54, no. 12, pp. 3186–3194, 2007. DOI: [10.1109/TED.2007.909030](https://doi.org/10.1109/TED.2007.909030).
- [265] M. Venn, *Tinytapeout.com*. [Online]. Available: <https://tinytapeout.com/>.
- [266] S. Bos, *Tapeout 4: Balanced ternary counter and signed binary radix converter*, 2023. [Online]. Available: https://github.com/aiunderstand/tt03p5-4-trit-balanced-ternary-counter-bt_signb_bt-radix-convertor.
- [267] S. Hassoun and T. Sasao, Eds., *Logic Synthesis and Verification*. Springer New York, 2001, ISBN: 978-0-7923-7606-4.
- [268] Berkeley MVSIS research group, *MVSIS: Logic synthesis and verification*, 2002. [Online]. Available: ptolemy.berkeley.edu/projects/embedded/mvsis/mvlogic.html.
- [269] R. Rudell and A. Sangiovanni-Vincentelli, ‘Exact minimization of Multiple-Valued Functions for PLA optimization,’ in *The Best of ICCAD: 20 Years of Excellence in Computer-Aided Design*, A. Kuehlmann, Ed. Springer, 2003, pp. 205–216. DOI: [10.1007/978-1-4615-0292-0_16](https://doi.org/10.1007/978-1-4615-0292-0_16).
- [270] L. Lavagno, S. Malik, R. Brayton and A. Sangiovanni-Vincentelli, ‘MIS-MV: Optimization of Multi-Level Logic with Multiple-Values Inputs,’ in *IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, 1990, pp. 560–563. DOI: [10.1109/ICCAD.1990.129981](https://doi.org/10.1109/ICCAD.1990.129981).
- [271] Y. Jiang and R. Brayton, ‘Logic optimization and code generation for embedded control applications,’ in *International Symposium on Hardware/Software Codesign*, 2001, pp. 225–229. DOI: [10.1109/HSC.2001.924680](https://doi.org/10.1109/HSC.2001.924680).

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [272] Berkeley Logic Synthesis and Verification Group, *ABC: A system for sequential synthesis and verification*, 2005. [Online]. Available: <https://people.eecs.berkeley.edu/~alanmi/abc>.
- [273] R. Brayton and A. Mishchenko, ‘ABC: An academic industrial-strength verification tool,’ in *Computer Aided Verification*, T. Touili, B. Cook and P. Jackson, Eds., Springer, 2010, pp. 24–40, ISBN: 978-3-642-14295-6.
- [274] S. Lin, Y.-B. Kim and F. Lombardi, ‘CNTFET-based design of ternary logic gates and arithmetic circuits,’ *IEEE Transactions on Nanotechnology*, vol. 10, no. 2, pp. 217–225, 2011. DOI: [10.1109/TNANO.2009.2036845](https://doi.org/10.1109/TNANO.2009.2036845).
- [275] C. Vudadha, A. Surya, S. Agrawal and M. B. Srinivas, ‘Synthesis of ternary logic circuits using 2:1 multiplexers,’ *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4313–4325, 2018. DOI: [10.1109/TCSI.2018.2838258](https://doi.org/10.1109/TCSI.2018.2838258).
- [276] M. Lin, Q. Han, W. Luo, X. Wang, J. Chen and W. Lyu, ‘A ternary memristor full adder based on literal operation and module operation,’ *International Journal of Circuit Theory and Applications*, vol. 50, no. 8, pp. 2932–2940, 2022. DOI: [10.1002/cta.3287](https://doi.org/10.1002/cta.3287).
- [277] A. Paugh, ‘Application of binary devices and Boolean algebra to the realisation of 3-valued logic circuits,’ *Proceedings of the Institution of Electrical Engineers*, vol. 114, 335–338(3), 3 Mar. 1967. DOI: [10.1049/piee.1967.0072](https://doi.org/10.1049/piee.1967.0072).
- [278] S. Lin, Y.-B. Kim and F. Lombardi, ‘A novel CNTFET-based ternary logic gate design,’ in *IEEE International Midwest Symposium on Circuits and Systems*, 2009, pp. 435–438. DOI: [10.1109/MWSCAS.2009.5236063](https://doi.org/10.1109/MWSCAS.2009.5236063).
- [279] S.-Y. Lee, S. Kim and S. Kang, ‘Ternary logic synthesis with modified Quine-McCluskey algorithm,’ in *International Symposium on Multiple-Valued Logic (ISMVL)*, 2019, pp. 158–163. DOI: [10.1109/ISMVL.2019.00035](https://doi.org/10.1109/ISMVL.2019.00035).
- [280] S. Kim, T. Lim and S. Kang, ‘An optimal gate design for the synthesis of ternary logic circuits,’ in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018, pp. 476–481. DOI: [10.1109/ASPDAC.2018.8297369](https://doi.org/10.1109/ASPDAC.2018.8297369).
- [281] M. Karnaugh, ‘The map method for synthesis of combinational logic circuits,’ *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, vol. 72, no. 5, pp. 593–599, 1953. DOI: [10.1109/TCE.1953.6371932](https://doi.org/10.1109/TCE.1953.6371932).
- [282] H. N. Risto, ‘A study of CNTFET implementations for ternary logic and data radix conversion,’ M.S. thesis, USN, Norway, 2020.
- [283] R. Brayton and S. Khatri, ‘Multi-Valued Logic synthesis,’ in *Proceedings Interna-*

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

108

-
- [284] L. Duret-Robert, *TernaryVerilog : A custom hardware description language*, 2020. [Online]. Available: <https://louis-dr.github.io/ternaryverilog.html>.
 - [285] M. M. Shulaker, G. Hills, N. Patil *et al.*, ‘Carbon Nanotube Computer,’ *Nature*, vol. 501, no. 7468, pp. 526–530, Sep. 2013. DOI: 10.1038/nature12502.
 - [286] H. Wang, P. Wei, Y. Li *et al.*, ‘Tuning the threshold voltage of carbon nanotube transistors by n-type molecular doping for robust and flexible complementary circuits,’ *Proceedings of the National Academy of Sciences*, vol. 111, no. 13, pp. 4776–4781, 2014. DOI: 10.1073/pnas.1320045111.
 - [287] S. Gadgil, G. N. Sandesh and chetan Kumar V, ‘Design and Implementation of a CNTFET-Based Ternary Logic Processor,’ Mar. 2023. DOI: 10.36227/techrxiv.22259437.v1.
 - [288] D. Patterson, ‘Reduced instruction set computers then and now,’ *Computer*, vol. 50, no. 12, pp. 10–12, Dec. 2017. DOI: 10.1109/MC.2017.4451206.
 - [289] A. Waterman and K. Asanović, *The RISC-V instruction set manual volume i: Unprivileged ISA*, 2019. [Online]. Available: <https://riscv.org/wp-content/uploads/2019/06/riscv-spec.pdf>.
 - [290] F. Pelletier and A. Hartline, ‘Ternary Exclusive OR,’ *Logic Journal of the IGPL*, vol. 16, no. 1, pp. 75–83, 2008. DOI: 10.1093/jigpal/jzm027.
 - [291] S. Bos, *Tapeout 3: Balanced ternary calculator*, 2023. [Online]. Available: <https://github.com/aiunderstand/tt03-balanced-ternary-calculator>.
 - [292] S. Bos, *1 trit balanced ternary ALU with MIN, MAX, STI, ADD, MULTIPLY gates*, 2022. [Online]. Available: <https://youtu.be/ApnVnEOL-ng>.
 - [293] S. Bos, *How to build a balanced ternary calculator chip using MRCS*, 2023. [Online]. Available: <https://youtu.be/-DzVKAxmSQ0>.
 - [294] Intel Corporation, *Intel 64 and IA-32 architectures software developer’s manual: Instruction set reference, a-z*, 2023. [Online]. Available: <https://cdrdv2-public.intel.com/774492/325383-sdm-vol-2abcd.pdf>.
 - [295] P. A. Samet, ‘A note on radix conversion for integers,’ *Software: Practice and Experience*, vol. 1, no. 1, pp. 93–96, 1971. DOI: 10.1002/spe.4380010109.
 - [296] S. Bos, *Tapeout 2: Binary encoded ternary to unary encoded ternary radix converter and comparator*, 2022. [Online]. Available: <https://github.com/aiunderstand/tt02-async-binary-ternary-convert-compare>.
 - [297] Fu-Qiang Li, M. Morisue and T. Ogata, ‘A proposal of josephson binary-to-ternary

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Bibliography

- [298] M. Shahangian, S. A. Hosseini and S. H. Pishgar Komleh, ‘Design of a multi-digit binary-to-ternary converter based on CNTFETs,’ *Circuits, Systems, and Signal Processing*, vol. 38, no. 6, pp. 2544–2563, Jun. 2019. DOI: 10.1007/s00034-018-0977-3.
- [299] T. Dhar, K. Kunal, Y. Li *et al.*, ‘ALIGN: A system for automating analog layout,’ *IEEE Design & Test*, vol. 38, no. 2, pp. 8–18, 2021. DOI: 10.1109/MDAT.2020.3042177.
- [300] D. Antoniadis, A. Mifsud, P. Feng and T. G. Constandinou, ‘An open-source RRAM compiler,’ in *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*, 2022, pp. 465–469. DOI: 10.1109/NEWCAS52662.2022.9842222.
- [301] B. Murmann and B. Hoefflinger, ‘The thirties,’ in *NANO-CHIPS 2030: On-Chip AI for an Efficient Data-Driven World*, B. Murmann and B. Hoefflinger, Eds. Cham: Springer International Publishing, 2020, pp. 577–583. DOI: 10.1007/978-3-030-18338-7_30.
- [302] WinChipHead, *32-bit general-purpose RISC-V MCU-CH32V003*, 2022. [Online]. Available: <https://github.com/openwch/ch32v003>.
- [303] S. Lefebvre, *ICE-V Dual*, 2021. [Online]. Available: <https://github.com/sylefeb/Silice/blob/master/projects/ice-v/IceVDual.md>.
- [304] G. Cowan, R. Melville and Y. Tsividis, ‘A VLSI analog computer/digital computer accelerator,’ *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 42–53, 2006. DOI: 10.1109/JSSC.2005.858618.
- [305] M. M. Sabry Aly, T. F. Wu, A. Bartolo *et al.*, ‘The n3xt approach to energy-efficient abundant-data computing,’ *Proceedings of the IEEE*, vol. 107, no. 1, pp. 19–48, 2019. DOI: 10.1109/JPROC.2018.2882603.
- [306] A. Shabarshin, *3-nity alpha: 9-trit ternary computer architecture*, 2004. [Online]. Available: <http://ternary.info/wiki/index.php?n=Alpha.About>.
- [307] A. Shabarshin, *Shared silicon: Binary, ternary, quaternary logic tapeout*, 2015. [Online]. Available: <https://hackaday.io/project/11779-shared-silicon>.
- [308] J. Connelly, *Ternary computing testbed 3 trit computer architecture*, 2008. [Online]. Available: <http://xyzzy.freeshell.org/trinary/CPE%20Report%20-%20Ternary%20Computing%20Testbed%20-%20RC6a.pdf>.
- [309] V. Lofgren, *Tunguska*, 2008. [Online]. Available: <https://tunguska.sourceforge.net/>.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)Available: <https://github.com/SBTCVM/SBTCVM-Gen2-9>.

- [311] D. Jones, *Trillium*, 2016. [Online]. Available: <https://homepage.cs.uiowa.edu/~dwjones/ternary/trillium.shtml>.

110

-
- [312] D. Jones, *Binary Coded Ternary and its inverse*, 2016. [Online]. Available: <https://homepage.cs.uiowa.edu/~dwjones/ternary/bct.shtml>.
- [313] D. V. Sokolov, *Triador: 3-trit balanced ternary computer architecture*, 2017. [Online]. Available: <https://hackaday.io/project/28579-homebrew-ternary-computer>.
- [314] L. Duret-Robert, *Simple as possible (SAP) one 9-trit*, 2019. [Online]. Available: <https://louis-dr.github.io/sap1.html>.
- [315] L. Duret-Robert, *CMOS implementation and analysis of Ternary Arithmetic Logic Unit*, 2019. [Online]. Available: <https://louis-dr.github.io/ternalu3.html>.
- [316] C. D. Mauro, *3isa: 20-trit ternary computer architecture*, 2019. [Online]. Available: <https://www.youtube.com/watch?v=ohhkcvuTL5k>.
- [317] C. L. Rosa, *24-trit computer project 5500fp*, 2019. [Online]. Available: <https://www.facebook.com/profile.php?id=100065548687080>.
- [318] S. Bos, *Base-base pair bruteforce solver*, 2020. [Online]. Available: <https://github.com/aiunderstand/Base-BaseIntApprox-PairSolver>.
- [319] N. J. A. Sloane and R. K. Guy, *Denominators of convergents to log_2 3 (formerly m1428)*, 1995. [Online]. Available: <https://oeis.org/A005664>.
- [320] D. Wust, D. Fey and J. Knödtel, ‘A programmable ternary CPU using hybrid CMOS/memristor circuits,’ *International Journal of Parallel, Emergent and Distributed Systems*, vol. 33, no. 4, pp. 387–407, 2018. DOI: 10.1080/17445760.2017.1422251.
- [321] Z. Liu, T. Pan, S. Jia and U. Wang, ‘Design of a novel ternary SRAM sense amplifier using CNFET,’ in *IEEE International Conference on ASIC (ASICON)*, 2017, pp. 207–210. DOI: 10.1109/ASICON.2017.8252448.
- [322] Y. Miyasaka, A. Mishchenko, J. Wawrzynek and N. J. Fraser, ‘Synthesizing a class of practical Boolean functions using truth tables,’ *Proceedings of the International Workshop on Logic & Synthesis, IWLS 2022*, 2022. [Online]. Available: https://people.eecs.berkeley.edu/~alanmi/publications/2022/iwls22_reo.pdf.
- [323] J. Ko, K. Park, S. Yong, T. Jeong, T. H. Kim and T. Song, ‘An optimal design methodology of ternary logic in Iso-device ternary CMOS,’ in *IEEE International Symposium on Multiple-Valued Logic (ISMVL)*, 2021, pp. 189–194. DOI: 10.1109/

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [324] A. C. Seabaugh and M. A. Reed, ‘Chapter 11 – Resonant-tunneling transistors, III Heterostructures and Quantum Devices,’ in *Heterostructures and Quantum Devices*, vol. 24, Elsevier, 1994, pp. 351–383. DOI: 10.1016/B978-0-12-234124-3.50016-1.
- [325] M. D. Miller and M. A. Thornton, ‘Functional representations,’ in *Multiple Valued Logic: Concepts and Representations*. Cham: Springer, 2008, pp. 43–67. DOI: 10.1007/978-3-031-79779-8_3.

111

Bibliography

- [326] J. McClellan, R. Schafer and M. Yoder, *Signal Processing First*. Pearson Prentice Hall, 2003, ISBN: 0-13-120265-0.
- [327] J. K. Saini, A. Srinivasulu and R. Kumawat, ‘Fast and energy efficient full adder circuit using 14 CNFETs,’ *Solid State Electronics Letters*, vol. 2, pp. 67–78, 2020. DOI: 10.1016/j.ssel.2020.09.002.
- [328] T.-D. Ene and J. E. Stine, ‘Point-targeted sparseness and ling transforms on parallel prefix adder trees,’ in *IEEE Symposium on Computer Arithmetic (ARITH)*, 2022, pp. 68–75. DOI: 10.1109/ARITH54963.2022.00021.
- [329] R. Tocci, N. Widmer and G. Moss, *Digital Systems: Principles and Applications*. Pearson, 2017, ISBN: 1292162007.
- [330] S. Kim, S.-Y. Lee, S. Park and S. Kang, ‘Design of quad-edge-triggered sequential logic circuits for ternary logic,’ in *IEEE International Symposium on Multiple-Valued Logic (ISMVL)*, 2019, pp. 37–42. DOI: 10.1109/ISMVL.2019.00015.
- [331] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th. USA: Addison-Wesley Publishing Company, 2010, ISBN: 0321547748.
- [332] R. F. Mirzaee and N. Farahani, *Design of a Ternary Edge-Triggered D Flip-Flap-Flop for Multiple-Valued Sequential Logic*, 2016. arXiv: 1609.03897.
- [333] S. Bos, *Mixed Radix Converter app*, 2022. [Online]. Available: <https://github.com/aiunderstand/ternary-workbench-unity/tree/master/MixedRadixConvert>
- [334] S. Bos, *Tapeout 1: 4-bit tristate loadable binary counter*, 2022. [Online]. Available: <https://github.com/aiunderstand/tt02-4bit-tristate-loadable-counter>.
- [335] Z. Han, ‘The Power-Delay Product and its implication to CMOS inverter,’ *Journal of Physics: Conference Series*, vol. 1754, no. 1, p. 012131, Feb. 2021. DOI: 10.1088/1742-6596/1754/1/012131.
- [336] M. Flynn, P. Hung and K. Rudd, ‘Deep submicron microprocessor design issues,’ *IEEE Micro*, vol. 19, no. 4, pp. 11–22, 1999. DOI: 10.1109/40.782563.
- [337] C. M. University, *Cmos power consumption*, 2003. [Online]. Available: <https://course.ece.cmu.edu/~ece322/LECTURES/Lecture13/Lecture13.03.pdf>.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

- [339] Unity, *Unity API: Application.persistentdatapath*, 2022. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>.
- [340] S. Bos, *How to build a balanced ternary calculator chip using MRCS*, 2023. [Online]. Available: <https://www.youtube.com/watch?v=-DzVKAxmSQ0>.

112

uMemristorToolbox: Open source framework to control memristors in Unity for ternary applications

Not available online due to publisher restrictions

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

113

Automated synthesis of netlists for ternary valued n-ary logic functions in CNTFET circuits

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Appendix G Additional material

Table G.4: Overview of useful arity-3 building blocks

Hepta Index	Name/Alias	Radix	Comment
KKKK00Z00	2:1 MUX	2	FE D-LATCH (feedback to B)
Z00K00KKK	2:1 MUX	2	RE D-LATCH (feedback to B)
K0200020K	SUM	2	XOR
ZKKK00K00	CARRY	2	
ZZZZKKZKK	MAX	2	OR
K00000000	MIN	2	AND
PPPPPZD0	2:1 MUX	3	FE D-LATCH (feedback to B)
ZD0PPPPP	2:1 MUX	3	RE D-LATCH (feedback to B)
ZD0DDDPBP	2:1 TRIMUX	3	Tristate with zero, not HZ
B7P7PB7	SUM	3	
XRDRDCDC9	CARRY	3	
ZZZZRRZRP	MAX	3	
PC0CC0000	MIN	3	

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

184

G.15 Combinatorial and sequential building blocks

G.15 Combinatorial and sequential building blocks

In this section ternary and mixed-radix combinatorial and sequential building blocks are discussed that were developed in this thesis work. These HSPICE and FPGA verified building blocks are essential for building a ternary computer.

Ternary data-latch

Figure G.10: 28T gated balanced ternary d-latch based on 2:1 MUX

Figure G.11: 46T gated balanced ternary d-latch based on NMIN

The latch is one of the most important memory elements. In Fig. G.10 a 28T level-controlled balanced ternary data-latch (d-latch) implementation is shown that was presented in paper F. The d-latch is effectively a 2:1 MUX with feedback from the output to

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

exist many other types of d-latch designs such as cross-coupled inverters with transmission gates or gated Set-Reset latches using 4 NAND gates. These are often found in high density binary SRAM cells. The same topologies can be used for ternary SRAM [321], [332]. A NAND (NMIN) ternary d-latch shown in Fig. G.11 costs 46T when made with MRCS and is not efficient. Both the NMIN and MUX d-latch design cost many more transistors/area than a 8T d-latch design based on two 2T cross-coupled STI's and two 2T transmission gates [149]. Transistor count is not always the most important metric. The 20T ternary SRAM made with CNTFET reported in [321] has 85% better performance and PDP compared to ternary 6 transistor, 2 capacitor (6T2C) DRAM also made with CNTFET.

185

Appendix G Additional material

Ternary data-flip-flop

Figure G.12: 54T rising-edge master-slave configuration balanced ternary d-flip-flop

Figure G.13: 52T rising-edge master-slave configuration unbalanced ternary d-flip-flop

A plethora of binary flip-flop designs exist[331]. While the latch is a level controlled memory element, the flip-flop is designed to be edge-controlled. Contrary to binary's bi-stable flip-flops ternary's flip-flops are tri-stable. Sometimes tri-stable flip-flops are called flip-flap-flops [332]. This term should be avoided as it is confusing and the name becomes rather grotesque with higher radices. Ternary flip-flops can be either binary clocked or ternary clocked. This is important as in modern (synchronous) computers the clock tree network (CTN) is the always-on backbone that can consumes 30% of the CPU power budget [8]. The rising-edge master-slave configuration of a MUX based ternary flip-flop is discussed in paper F and shown in Fig. G.12. This configuration consists of two ternary d-latches with a binary inverter. The inverter can be integrated (reducing 2T) by flipping the heptavintimal index of the second latch from 0tPPPPPZD0 to 0tZD0PPPPP. By reversing the latches the flip-flop becomes either a rising-edge or falling-edge flip-flop. Note that in Fig. G.12 a balanced ternary version of the D-flip-flop is shown while in Fig. G.13 an unbalanced ternary version of the d-flip-flop is shown. Both have identical

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Ternary data-flip-flop with double/quad data rate

The d-flip-flop design in Fig. G.12 is a single data rate (SDR) or single edge d-flip-flop. It only transitions with a rising-edge. If tighter timing is permissible, then double data rate (DDR) d-flip-flops or both rising and falling edge triggered d-flip-flops make sense. A novel 76T DDR ternary d-flip-flop design is shown in Fig. G.14 which uses both edges of the binary clock. The design is based on the latches $0pPPPZD0PPP$ and $0pZD0PPPZD0$. With a ternary clock the transitions can be increased to 4 edges. Quad data rate or quad edge triggered d-flip-flops require even tighter timing. The power benefits of QDD memory versus SDD is substantial as the voltage swings are smaller in both the CDN and d-flip-flops. Kim et al. [330] show that QDD ternary flip-flops are 31% more efficient

186

G.15 Combinatorial and sequential building blocks

Figure G.14: 76T DDR master-slave configuration balanced ternary d-flip-flop

Figure G.15: 110T QDR master-slave configuration balanced ternary d-flip-flop

than SDD ternary flip-flop (only slightly higher delay) while the clock tree consumes 75% less power. A novel MUX-based 110T QDR ternary flip-flop design is shown in Fig. G.15. This design is based on the latches $0tPPPZD0PPP$ and $0tZD0PPPZD0$ and one ternary 2:1 MUX. This MUX has the same heptavintimal index but has no feedback making it a combinatorial block. The top latch is responsible (active) for the edges -1 to 0 and 1 to 0 while the bottom latch is responsible for the edges 0 to 1 and 0 to -1. The wiring is important as the MUX responds to the active latch only during an edge. During level it "listens" to the inactive latch thus ignoring any changes.

Ternary register

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

with MUX-based ternary d-flip-flops shown in Fig.G.12. The *Read* flag is a binary signal while the output is ternary, thus requiring the usage of a ternary logic gate. Enabling the read flag allows the output to reflect the truth table output, else a constant zero is output. This is useful in combination with a DESELECT component ($0tVP0$) allowing multiple registers to be connected without using a transmission gate and bus architecture. The register design is thus a pure logic gate based design. Transmission gates however are far more efficient, costing just 2T instead of 16T for the *Read* flag. Transmission gates seem a better fit for this functionality in combination with higher radix circuits as transmission gates signals are analog in nature. It should be noted that Kim et al. [280] report that transmission gates "have worse power, speed, and noise margin than static gates for ternary". The ternary register is an example of a mixed-radix design, combining various binary and ternary signals which reduces the transistor count compared to a pure ternary implementation.

187

Appendix G Additional material

Figure G.16: 80T balanced ternary register

Ternary ROM/RAM

Registers give a blueprint to construct larger memory elements such as blocks of RAM or ROM. Typically application code and immutable data (constants) is stored in ROM while processed data is put in RAM. ROM is often a non-volatile memory array such as FLASH while processed data is stored in SRAM or DRAM and is lost after a power cycle. RAM/ROM have the same interface as registers but feature a DEMUX/MUX to select individual (blocks of) memory elements. The building blocks of both ROM and RAM are clusters of balanced ternary d-flip-flops and is shown in Fig. G.17 .

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure G.17: RAM-3 implementation with three d-flip-flops

A 2x1 block of RAM is shown in Fig. G.18. The 2x1 RAM has 2-trit addresses meaning thus can refer to 9 unique addresses for both reading and writing. Each address refers to a single ternary d-flip-flops. By adding another 2x1 block of RAM in parallel a 2x2 RAM block can be made, thus expanding the content while keeping the address width the same. Parallel read actions are made possible by adding another MUX with two 2-trit register-source addresses (RsAddr and RsAddr2). Parallel write actions are slightly more complex as two write actions might want to update the same register with different data. In MRCS the RAM/ROM content is *uninitialized* at first and needs to be reset

188

G.15 Combinatorial and sequential building blocks

Figure G.18: Ternary ROM/RAM. Implementation of (DE)MUX are shown in Subcomponents

(for instance to 0_3 , 1_3 or 0_2). This is similar to actual behavior of physical memory elements which exhibit unknown states at initialization. Initializing or programming the RAM/ROM can be accomplished manually or automated by reusing the *verify component*

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Ternary full adder

The addition instruction is the cornerstone of most CPU architectures [2], [328]. It is one of the basic arithmetic operations next to subtraction, multiplication and division. These four operations are not uniformly used as they are often reduced to addition and shift operations. Subtraction is addition with one input inverted (2's complement), multiplication is repeated addition and division is repeated subtraction (complete algorithms for all three operations are slightly more complex, see [2]). Even the program counter uses an adder with a constant (such as a hardwired 1) to compute the next instruction address from the present instruction address. Statistics vary, but in [2] the most common instruction of RISCV CPU's is the ADD instruction. Optimizing the adder result in massive, system-wide performance boost. For this reason the binary adder is historically well researched and is still being improved [327], [328].

The recent large scale survey on ternary full adders (TFA) by Nemati et al. [143] show that a multitude of adder designs exist including many based on multi-threshold CNTFET. Surprisingly, 90% of the reported ternary adders in literature used unbalanced ternary

189

Appendix G Additional material

Figure G.19: 110T BTA design with SUM-based CARRY

encoding. Consider for example the ternary half adders (THA) consisting of a SUM and CONS (carry) gate. The 2-ary unbalanced SUM circuit (0tB7P) is 31T and CONS (0tC90) is 10T plus 8T for shared NTI/PTI inverters, totalling 49T. For balanced ternary SUM (0t7PB) is 32T, CONS (0tRDC) is 10T plus 8T for shared inverters, totalling 50T.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

The survey papers by Nemati et al. [143] concludes: "A TFA with faster operation, lower power consumption, and fewer transistors is needed to be considered a potential rival for the binary counterparts". A similar conclusion is found in a survey by Etiemble [225]. Although full adder designs certainly exist with competitive transistor count [202], they have not been demonstrated to be competitive in direct comparison according to PPAC metrics. A survey on binary full adders usings CNTFET [327] show that 14T is possible with transmission gate logic (TG) compared to the well known 28T design using static logic. As mentioned in Chapter 2, for fair comparison to ternary similar logic styles, functionality, resolution, input pattern, etc should be used. Only then will PPAC comparison makes sense. This is unfortunately rarely done in survey papers. For example, the balanced ternary SUM gate made with MRCS is 32T which is 4x larger than the 8T SUM (CMOS XOR) gate in binary. However, the difference in transistor count becomes small when compensating for identical features. The binary 2-bit signed addition circuit requires 2x 2-bit input to cover the range of 2x1 trit input, additional circuitry, wiring, and a add/sub functionality pin. A straightforward implementation would require 3 SUM gates and a AND (carry) for a total of 30T. Even when compensating for theoretical identical resolution as the 2 bit signed binary adder can compute 3 more states (-3,-4 and +2), the difference in transistor count is competitive.

190

G.15 Combinatorial and sequential building blocks

Kim et al. [155] shows a 118T balanced ternary full adder which is replicated in [282]. Just like in a logical level optimized 28T binary full adder design, a TFA with carry that depend on SUM output can be constructed (see Fig. G.19). This design cost 110T and is thus 8T smaller. The SUM components are made with 0t7PB and carry with 0tRR99DDDXCC. Unclear is if this design has been reported earlier.

Ternary asynchronous counter

Figure G.20: Balanced ternary ripple counter

Figure G.21: 2-trit balanced ternary ripple counter

With a memory and adder block another common building block can be constructed: the

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

instructions it fulfills a role as program counter (PC). Two classes of counters exist, asynchronous or ripple counters and synchronous or parallel counters [329]. Asynchronous balanced ternary counters can be constructed with a d-flip-flop such as Fig. G.12 and INCREMENT (0t7). Contrary to binary counters using JK-flip-flops ternary counters don't toggle between two states such that the output is reusable as a binary clock signal. By adding a NTI (0t2) to detect the overflow, multiple ternary ripple counters can be stacked (see Fig. G.20 and Fig. G.21).

Ternary synchronous program counter

In [260] binary and balanced ternary synchronous counters are shown which were made with MRCS. The various designs are verified with HSPICE simulations. These counters are classified as up/down counters and can be loaded to a specific count value. This makes them usable as a program counter (PC). The PC normally increments linearly but some instructions jump to other instructions at a different address (count value), creating the data-driven flows needed for non-trivial computations. The design of the PC consists of 3 components: ADDER (0t7PB) to count up/down or not count, MUX to choose between the adder or load input and a d-flip-flop (see Fig. G.12) to store the input. In Fig. G.22 a single balanced ternary counter is shown. A CONSENSUS (0tRDC) gate is used between the counters to detect the positive or negative overflow, depending on the direction. This

191

Appendix G Additional material

design is another example of a mixed-radix design where some signals are binary and some are balanced ternary resulting in a smaller transistor count than binary encoded ternary or pure ternary.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Both the 6-bit binary counter in [260] and 4-trit balanced ternary counter in Fig. G.23 have been submitted for tape-out [266], [334]. These counters are designed to have identical features. With CNTFETs the 6-bit binary counter needs 542 transistors while the 4-trit ternary needs 8 less, 534 transistors. Less transistors are needed for ternary while the resolution of 4-trits being 81 is higher than the resolution of 6-bits (= 64). Note that no radix economy compensation is applied as the resolution is more or less comparable. A slight compensation to have identical resolution would benefit ternary even more but is purely theoretical since it would require non-discrete devices. The small transistor/die area advantage for ternary increases with higher trit comparisons since the designs are compoundable.

Although in this comparison ternary has a lower transistor count, the average power consumption and delay and thus the PDP is much worse compared to binary. The binary design used $18 \mu W$ for a basic testbench (see [260]) while the ternary design used $137 \mu W$. This big difference is the result of the synthesis method discussed earlier. The $\frac{VDD}{2}$ state is made by voltage division and consumes a disproportional amount of current. In [155] Kim et al. show that this state consumes 266.36 nW while logical -1 and logical 1 consume 0.15 nW and 0.31 nW respectively. In the same work they propose to use the body effect to reduce the static power consumption of the middle voltage. They improved the power consumption from 266.36 to $3.57\mu W$. New simulations are needed to ascertain if the PDP is below the binary design with this improvement.

192

G.15 Combinatorial and sequential building blocks

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure G.23: 4-trit synchronous balanced ternary tri-directional loadable program counter

193

Appendix G Additional material

G.16 Subcomponents

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure G.26: DEMUX level 2 implementation, part of Fig. G.18

Figure G.27: DEMUX level 1 implementation, part of Fig. G.18

Figure G.28: XOR-3 implementation, part of Fig. 4.9. The XOR gate is made with binary $0t20K$.

Figure G.29: BHA-3 implementation part of Fig. 4.9. The BHA gate is made with binary $0t20K$ for the sum and binary $0tK00$ for the carry.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

0/5DP.

Figure G.31: The 170T 4-bit unsigned binary to 4-trit balanced ternary radix converter in paper E and part of Fig. 4.9

195

Appendix G Additional material

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure G.32: A novel 139T 3-trit balanced ternary to 4-bit 2's complement signed binary radix converter

196

G.17 Online radix conversion tool

G.17 Online radix conversion tool

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Figure G.33: User interface of the online radix converter tool.

During the development of MRCS and the radix conversion circuits many conversions were needed. For ternary to binary and the inverse a very fast software converter can be found [312]. A more general approach between all possible radices is found in [295]. No all-in-one and browser-based radix converter could be found that was able to convert between radix-2, radix-3 and radix-10 in both signed and unsigned encoding. This lead to the development of the open source radix converter tool [333]. It was made with Unity WebGL and is hosted on TernaryResearch.com/mixed-radix-converter/. It is planned to add heptavintimal, octal, hexadecimal and base-64 conversion in a future version.

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

198

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

TERNARY 2025. TERNARY VLSI WORKFLOW

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

T

Tape-outs

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Citations (1)

[References \(306\)](#)**REBEL-6: A 32-trit balanced ternary instruction set architecture with R2R compiler pipeline for C**[Conference Paper](#)

Jun 2025

Steven Bos . Vetle Bodahl . Ole Christian Moholth . Henning Gundersen

[View](#) [Show abstract](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)

Recommended publications

[Discover more](#)[Conference Paper](#) [Full-text available](#)

uMemristorToolbox: Open source framework to control memristors in Unity for ternary applications

November 2020

Steven Bos · Henning Gundersen · Filippo Sanfilippo

[View full-text](#)[Conference Paper](#)

Post-Binary Robotics: Using Memristors With Ternary States for Robotics Control

September 2020

Steven Bos · Henning Gundersen · Julian Breivold Nilsen

[Read more](#)[Article](#)

A data-centric chip design agent framework for Verilog code generation

April 2025 · ACM Transactions on Design Automation of Electronic Systems

Kaiyan Chang · Wenlong Zhu · Kun Wang · [...] · Ying Wang

Recent advances in large language models (LLMs) have demonstrated significant potential for automated hardware description language (HDL) code generation from high-level specifications. However, two critical challenges limit further progress in this domain: the scarcity of quality Verilog training data and the inability of current approaches to generate RTL code optimized for power, performance, ... [\[Show full abstract\]](#)

[Read more](#)[Conference Paper](#)

Ndr Effects in a Locally-Active Memristor Induce Small-Signal Amplification in a Simple Cell

June 2025

Alon Ascoli · Emanuele Gemo · Fernando Corinto · [...] · Leon O. Chua

[Read more](#)**Company**[About us](#)
[News](#)**Support**[Help Center](#)**Business solutions**[Advertising](#)
[Recruiting](#)

[Download full-text PDF](#)[Download citation](#)[Copy link](#)[Terms of use](#) · [Privacy](#) · [Copyright](#) · [Imprint](#) · [Content preferences](#)