

**POST-SYNTHESIS MANIPULATION OF  
FIELD-COUPLED NANOCOMPUTING CIRCUITS  
FOR FUNDAMENTAL ENERGY REDUCTION**

JEFERSON FIGUEIREDO CHAVES

**POST-SYNTHESIS MANIPULATION OF  
FIELD-COUPLED NANOCOMPUTING CIRCUITS  
FOR FUNDAMENTAL ENERGY REDUCTION**

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

ADVISOR: OMAR PARANAÍBA VILELA NETO

Belo Horizonte

August 2020

© 2020, Jeferson Figueiredo Chaves.  
Todos os direitos reservados.

Chaves, Jeferson Figueiredo.

C512p

Post-synthesis manipulation of field-coupled nanocomputing circuits for fundamental energy reduction / Jeferson Figueiredo Chaves.- 2020.  
xxii, 88 f. il.

Orientador: Omar Paranaíba Vilela Neto.

Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: f.79-88.

1. Sistemas Digitais – Teses. 2. Arquitetura de computador – Teses. 3. Nanocomputação – Teses. 4. Computação Reversível – Teses. I. Vilela Neto, Omar Paranaíba. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6\*21(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa  
CRB 6ª Região nº 1510



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

Post-Synthesis Manipulation of Field-Coupled Nanocomputing Circuits for  
Fundamental Energy Reduction

**JEFERSON FIGUEIREDO CHAVES**

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. OMAR PARANAÍBA VILELA NETO - Orientador  
Departamento de Ciência da Computação - UFMG

PROF. RENATO PEREZ RIBAS  
Departamento de Informática Aplicada - UFRGS

PROF. FRANK SILL TORRES  
Centro Aeroespacial Alemão - DLR

PROF. LUIZ FILIPE MENEZES VIEIRA  
Departamento de Ciência da Computação - UFMG

PROF. MARIO SÉRGIO FERREIRA ALVIM JÚNIOR  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 17 de Agosto de 2020.

*To my family: Ana Beatriz, Álvaro and Otávio.*

# Acknowledgments

Gostaria de agradecer a todos que contribuíram para a conclusão desta tese de doutorado.

Agradeço especialmente à minha família, Ana Beatriz, Álvaro e Otávio, pelo apoio e compreensão pelos inúmeros momentos de ausência.

Ao meu orientador, Professor Omar Paranaíba Vilela Neto, que contribuiu de forma significativa na minha formação de pesquisador. Agradeço pelas inúmeras oportunidades criadas, discussões e pelo constante direcionamento.

Aos co-autores dos inúmeros artigos publicados, agradeço pela colaboração e discussões. Em particular, agradeço especialmente aos mais recorrentes: Omar, Marco Antônio, Iago, Frank, Leandro.

Aos professores Renato Perez Ribas, Frank Sill Torres, Luiz Filipe Menezes Vieira, Mario Sérgio Ferreira Alvim Júnior pela participação na avaliação da tese e pelas valiosas contribuições.

Ao Centro Federal de Educação Tecnológica de Minas Gerais pelo liberação dos encargos didáticos para a dedicação exclusiva ao doutoramento.

Ao Programa de Pós-Graduação em Ciência da Computação da UFMG pelo alto padrão de qualidade e exigência.

Por fim, aos colegas do NANOCOMP pela amizade e pela excelente convivência no laboratório.

# Resumo

O avanço no desempenho de computadores é promovido em diversas frentes. Iniciativas como a implementação de algoritmos ótimos, paralelização de algoritmos e arquiteturas e, em um nível mais baixo, a Lei de Moore, têm garantido a evolução do desempenho. Essa última, em especial, já apresenta claros sinais de esgotamento impondo à indústria e à academia a busca de alternativas. Isso ocorre porque a geração de calor tem sido um dos principais entraves na continuidade da redução das dimensões dos dispositivos digitais. Landauer investigou a origem deste problema e demonstrou analiticamente em 1961 que a causa é o apagamento de informação que ocorre em cada porta lógica. O Princípio de Landauer, nome pelo qual este limite energético fundamental ficou conhecido, foi provado experimentalmente em 2012 indicando que de fato há um limite na escalabilidade energética para sistemas computacionais convencionais. Neste trabalho, são identificadas e exploradas oportunidades de redução dos limites energéticos fundamentais em circuitos digitais de dispositivos de acoplamento local de campo. As estratégias propostas de redução do apagamento de informação são baseadas em manipulações na temporização ou no leiaute dos circuitos após o processo de síntese lógica. Avaliando o impacto das alterações da temporização sobre um conjunto de benchmarks do estado-da-arte, foi observado que há um grande espaço de configurações para os circuitos. Neste caso há um compromisso entre vazão e energia onde a melhora de uma destas métricas implica na degradação da outra. Através da análise deste espaço de configurações foi possível indicar estratégias ótimas que permitem melhorar uma métrica e minimizar a degradação da outra. Especificamente, foi desenvolvido um algoritmo que, dada a rede lógica de um circuito, retorna a configuração que produz a energia dissipada mínima enquanto considerada uma restrição mínima de vazão. Também foi desenvolvido um algoritmo em que dada uma restrição máxima de energia, encontra a vazão máxima. Desta forma, um projetista pode encontrar a melhor configuração de temporização dado seus requisitos de projeto. Experimentos demonstraram que os algoritmos propostos superam o algoritmo do estado-da-arte em qualidade e quantidade de soluções, sendo capaz de dominar 56% das melhores soluções e gerar  $11\times$  mais

soluções em média. Outra contribuição deste trabalho é a proposta de um novo tipo de sistema parcialmente reversível. A estratégia de redução de perdas energéticas consiste na mudança de leiaute do circuito de forma a incorporar derivações (fanouts) em portas lógicas. Observa-se neste caso que os limites energéticos fundamentais podem ser reduzidos em média em 44% sem nenhum atraso adicional dos sinais lógicos. Caso atrasos não sejam um problema, a redução média pode alcançar até 77%. Por fim, foi realizada uma análise unificada dos sistemas resultantes de cada estratégia. Estes esforços apresentados criam novas perspectivas para o projeto de circuitos de dispositivos de acoplamento local de campo onde a eficiência energética é essencial.



# Abstract

Progress in computer performance is promoted on several fronts. Initiatives such as the implementation of optimal algorithms, parallelization of algorithms and architectures, and, at a lower level, Moore's Law, have guaranteed performance evolution. The latter, in particular, already shows clear signs of exhaustion, imposing on industry and academia the search for alternatives. The reason is heat generation, which has been one of the main obstacles to reducing digital devices' dimensions. Landauer investigated the origin of this problem and demonstrated analytically in 1961 that the cause is the erasure of information in each logic gate. This fundamental energy limit, named Landauer's Principle, was experimentally proven in 2012, indicating that there is indeed a limit on energy scalability for conventional computer systems. In this work, we introduce approaches to reduce those losses by applying post-synthesis methods. First, we propose a novel method to divide Partially Reversible Pipelined circuits into stages, computing the optimal energy vs. throughput tradeoff. Specifically, we develop an algorithm that, given the circuit, it returns the division that yields the minimum possible energy dissipation while considering a minimum throughput restriction. We also developed an optimal algorithm that, given the maximum energy restriction, finds the maximum possible throughput. Therefore, the designer can have the best partially reversible pipelined circuit configuration according to its project requirements. Computational experiments show that our algorithm outperforms the state-of-the-art algorithm from the literature in solution's quantity and quality, being able to dominate about 56% of its best solutions and to generate  $11\times$  more solutions, on average. The other strategy is a novel technique that identifies exploitable fan-outs and uses them to reduce energy losses in FCN circuits, thus enabling the design of new types of partially reversible systems. Moreover, we propose an algorithm that creates partially reversible systems while allowing the designer to choose between energy reduction with no effect on the delay or focus solely on energy and accepting a potential delay penalty. Simulation results for state-of-the-art benchmarks indicate an average reduction of the fundamental energy limit by 44% without affecting the delay. If delay is not the main

concern, the average reduction reaches even 77%. Finally, we present a unified analysis of both partially reversible systems. These efforts open new perspectives in designing FCN systems where energy efficiency is mandatory.

# List of Figures

1.1	Intel’s processors evolution [Cross, 2016]. . . . .	2
2.1	Potential energy landscape of bistable devices and system state (red dot). The example presents a system in "0" state and a barrier of at least $k_B T$ Joules between the two states. . . . .	8
2.2	An adiabatic RESTORE TO 1 operation. . . . .	9
2.3	Physical gate abstraction. Adapted from Anderson [2013]. . . . .	11
2.4	Three types of logic operations. The green color represents information flowing through a computational system. The red color shows irreversible information erasure. Irreversible process (a) presents two cycles, Reversible process case 1 (b) shows only one cycle and Reversible process case 2 (c) presents two cycles. . . . .	12
2.5	QCA: (a) A basic cell. (b) A cell fixed on <i>false</i> . (c) A cell fixed on <i>true</i> . (d) A wire. (e) An inverter gate. (f) A majority gate. . . . .	15
2.6	An example of Landauer clocking scheme controlling the polarization of cells in a circuit with two majority gates. Each curve represents the tunneling potential barriers of its respective zone. The circuit loses information in each clock cycle in the zones with majority gates (highlighted in red). . . .	16
2.7	An example of Bennett clocking scheme controlling the polarization of cells in a circuit with two MAJ gates. There are no information losses within the circuit since each erasure happens with an adjacent copy (highlighted in purple). . . . .	18
2.8	An example of a partially reversible pipeline with two stages, each of which with three Bennett zones. The dotted line represents the division between stages. Stage 2 loses information each clock cycle in its input’s cells since there are no adjacent copies (highlighted in red). . . . .	20

3.1	Wave base for Bennett’s clocking signals generation. QCADesigner uses the tunneling probability as clock signals rather than the tunneling potential barriers. The clocking phases <i>Switch</i> , <i>Hold</i> , <i>Release</i> and <i>Relax</i> are indicated by sw, hd, rs and rx, respectively. . . . .	23
3.2	Clocking signals for a circuit with 8 zones divided in 2 stages. . . . .	25
3.3	Validation of our Bennett clocking implementation. . . . .	26
3.4	4-to-1 multiplexer: (a) circuit layout and (b) functional simulation results. 2-bit adder: (c) circuit layout and (d) functional simulation results. . . . .	27
3.5	In-depth analysis of EPFL’s Adder benchmark . . . . .	29
3.6	In-depth analysis of EPFL’s Memory Controller benchmark . . . . .	30
3.7	In-depth analysis of EPFL’s Sin benchmark . . . . .	31
3.8	Netlist manipulation methods. . . . .	35
3.9	Section-splitter example. . . . .	38
3.10	Optimal Splitter Algorithm example. . . . .	41
3.11	In-depth analysis of <i>sin</i> benchmark: SSA vs. OSA. . . . .	45
4.1	Half-adder circuit. . . . .	50
4.2	Symbols and layouts of conventional as well as <i>n</i> -bits recycling QCA <i>AND</i> gates. . . . .	52
4.3	Application of proposed method for the half-adder circuit. . . . .	53
4.4	Half-adder circuit that exploits fan-outs with delay penalty. . . . .	55
4.5	Half-adder circuit QCA layouts. . . . .	58
4.6	Generations of chains in the depth-oriented strategy. . . . .	61
4.7	Generations of chains in the energy-oriented strategy. . . . .	61
4.8	Integration of fully reversible gates. . . . .	61
4.9	Circuit graphs resulting from the proposed strategies: the <i>i1</i> circuit’s case. . . . .	62
4.10	<i>sin</i> circuit’s evaluation. . . . .	63
5.1	Configuration space (Throughput-Energy-Depth) for EPFL’s Adder. . . . .	68

# List of Tables

3.1	Throughput and Energy evaluation results . . . . .	33
3.2	Energy reduction from using <i>lazy fanouts</i> . . . . .	37
3.3	Section-splitter algorithm <i>vs.</i> the naïve approach. . . . .	39
3.4	OSA <i>vs.</i> SSA. . . . .	46
4.1	Entropy calculations for gates in the half-adder of Fig. 4.1 . . . . .	51
4.2	Entropy calculations for gates in the half-adder circuit depicted in Fig. 4.3b	54
4.3	Entropy calculations for gates in the half-adder circuit depicted in Fig. 4.4	56
4.4	Energy values in $k_B$ Joules for half-adder circuits. . . . .	56
4.5	Technology and simulation parameters in the tool <i>QCADesigner-E</i> . . . . .	57
4.6	Benchmark results . . . . .	60
5.1	Configuration spaces . . . . .	69
5.2	Depth reduction effects in Landauer clocked circuits. . . . .	72
5.3	Depth reduction effects in partially reversible pipelines . . . . .	73

# Contents

Acknowledgments	v
Resumo	vi
Abstract	viii
List of Figures	x
List of Tables	xii
<b>1 Introduction</b>	<b>1</b>
1.1 Research Motivation . . . . .	2
1.2 Thesis Statement and Goals . . . . .	5
1.3 Contributions . . . . .	5
1.4 Thesis Organization . . . . .	6
<b>2 Background</b>	<b>7</b>
2.1 Irreversibility and Heat Generation in the Computing Process . . . . .	7
2.2 Logical Reversibility of Computation . . . . .	9
2.3 Gate Abstractions and Reversibility . . . . .	10
2.4 Field-Coupled Nanocomputing Circuits . . . . .	13
2.5 Clocking Systems . . . . .	14
2.6 Summary . . . . .	21
<b>3 Partial Reversibility via Clocking</b>	<b>22</b>
3.1 Partially Reversible Pipelines in QCADesigner . . . . .	22
3.2 Energy-Throughput Relations . . . . .	26
3.3 Algorithms . . . . .	34
3.3.1 Lazy Fanouts Netlist Manipulation Strategy . . . . .	35
3.3.2 The Section-Splitter Algorithm . . . . .	36

3.3.3	The Optimal Splitter Algorithm . . . . .	40
3.4	Summary . . . . .	47
<b>4</b>	<b>Partial Reversibility via Layout</b>	<b>48</b>
4.1	Energy Evaluation in FCN Circuits . . . . .	48
4.2	Energy Reduction by Layout Changes . . . . .	52
4.3	The Bit-Recycling Algorithm . . . . .	56
4.4	Summary . . . . .	63
<b>5</b>	<b>Towards a Unified Analysis</b>	<b>65</b>
5.1	Energy Evaluation . . . . .	65
5.2	Energy-Throughput-Depth Spaces . . . . .	66
5.3	Energy and Synthesis' Depth Reduction . . . . .	70
5.4	Summary . . . . .	71
<b>6</b>	<b>Conclusions</b>	<b>75</b>
6.1	Future Work . . . . .	76
	<b>Bibliography</b>	<b>77</b>

# Chapter 1

## Introduction

The semiconductor industry has always acted as one of the main players in computer performance evolution through Moore's Law support. Proposed by Gordon Moore in 1965, his law states that the number of transistors in integrated circuits doubles within a few years [Moore, 1965]. More than an empirical observation, this "law" absorbed other meanings that together became goals for the industry [Mollick, 2006]. Dennard et al. were the first to formally broaden these meanings by associating Moore's Law with transistor feature size, delay time, and power density reductions [Dennard et al., 1974].

In addition to these technical aspects, Moore's Law also embodied economic characteristics. As transistors shrunk and chips became faster, the market grew to not only enable the return from investment in research and development but also allowed the funding for next generation advancement [Cross, 2016]. Figure 1.1 quantitatively presents the evolution of these ideas and also reveals some interesting information. Although the number of transistors per chip continues to follow the same pattern, some aspects no longer improve, such as the clock frequency. When it comes to the monetary cost, the situation is even worse as each transistor has become more expensive.

Power dissipation deserves special attention, as it is considered one of the main obstacles to survival of Moore's Law [DeBenedictis, 2017]. The main reason for that is transistor miniaturization, which has led to reducing the channel-to-gate insulating layer to a few atoms. In a thinner layer, electron tunneling is facilitated and, consequently, leakage current increases. This scenario contributes to a decrease in energy efficiency and a less predictable operation of the device, i.e., the transistor deviates from the ideal model of an electronically controlled switch. Although alternatives to minimize this problem have already emerged, such as the FinFET or the stacked nanosheet FET, recent studies indicate that this device is reaching the end of its energy



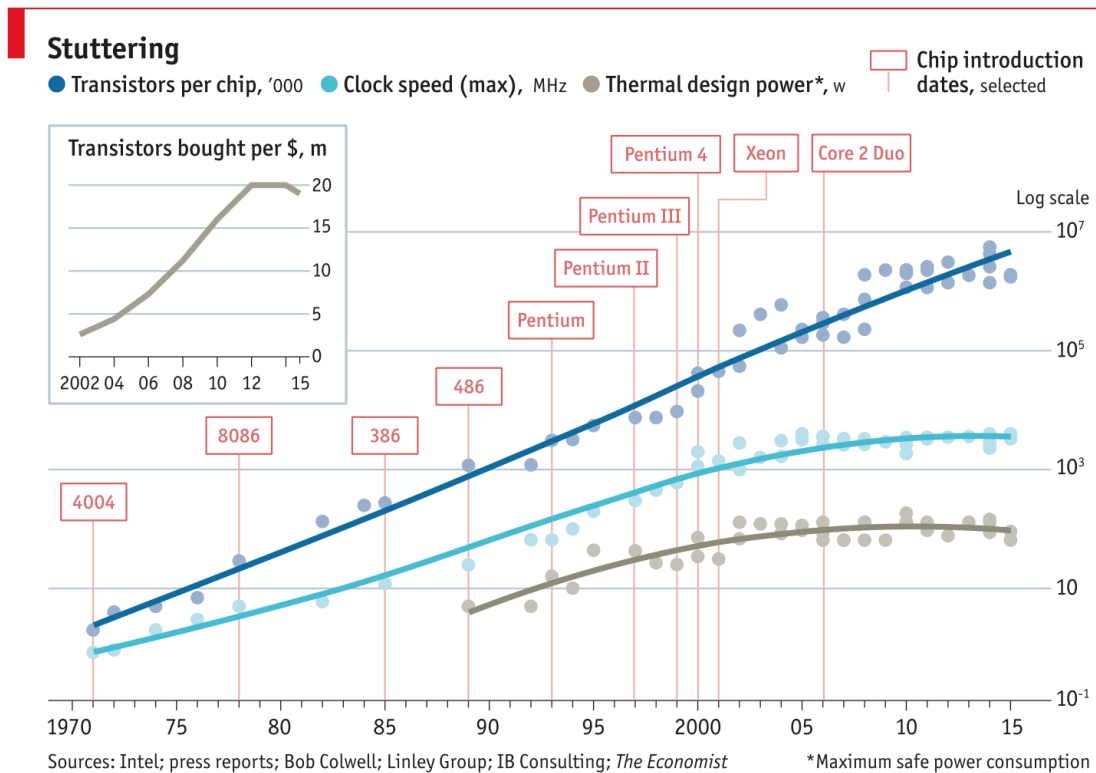


Figure 1.1: Intel's processors evolution [Cross, 2016].

improvement process [DeBenedictis, 2016; Ye et al., 2019].

## 1.1 Research Motivation

“The search for faster and more compact computing circuits leads directly to the question: What are the ultimate physical limitations on the progress in this direction?” [Landauer, 1961]. This visionary statement from Landauer reveals that he was already concerned with how far computing devices could reach. In his groundbreaking work, he showed that every irreversible computational process, i.e., one that results in information loss, dissipates a proportional amount of energy related to the Laws of Thermodynamics.

Landauer meant by Information a description at the lowest level of any physical artifact performing a specific computation. Despite the decades of debate about this energy-information relationship, Landauer's idea was experimentally verified in different physical artifacts, confirming a physical limit in irreversible computation [Bérut et al., 2012; Orlov et al., 2012; Hong et al., 2016; Lent et al., 2018]. These experiments

have also been causing a change of perspective regarding Information. It began to be recognized as a physical quantity [Parrondo et al., 2015; Anderson, 2017, 2018; Frank, 2018b; Lent et al., 2018; Lent, 2019]. Unfortunately, although Landauer’s principle dates all the way back to the 1960s, only recently these discussions have gained attention and visibility since the thermodynamic view of computers cannot be neglected anymore [Conte et al., 2019].

Landauer’s limit has severe implications. This ultimate energy limit is genuinely a lower bound exclusively related to the performed computation. It is independent from the physical quantity’s type used to encode bits, e.g., electrical, mechanical, magnetic, and so on. Even with an additional component in the overall energy related to those specific physical degrees of freedom, the total energy loss could not be less than the information loss. There is why we call it fundamental energy loss, i.e., it is unavoidable. Therefore, reducing information losses is mandatory to improve the system’s energetic scalability, considering that technological advances can reduce non-fundamental losses.

Landauer believed that irreversibility plays an essential and inevitable role in any computational process. It took a little more than a decade for his colleague Bennett to show that any irreversible computational process could be mapped onto a reversible structure, thus avoiding fundamental energy losses [Bennett, 1973]. Accurately, Bennett presented a natural computational process that worked according to his idea, the transcription of a DNA strand into RNA. Proposals for device implementations that could incorporate the same ideas were missing. The first contributions only came in 1982 from Fredkin and Toffoli’s work, who proposed electrical and mechanical systems capable of operating reversibly [Fredkin and Toffoli, 1982]. Their work also presented the first reversible logic gates and a ballistic computing model.

Fredkin and Toffoli’s systems involved many idealizations, which posed substantial engineering challenges in their construction. It was not until the 1990s that the first silicon-based proofs of concept emerged. A reversible logic family has been proposed and constructed, as well as reversible processors [Younis and Knight, 1993; Vieri et al., 1998]. Despite these significant innovations in building reversible systems, at the same time, the industry has also made progress with its conventional (irreversible) computing model. This scenario made funding scarce as the reversible approach would involve a complete change in the industry and processes, so few new proposals for physical implementation appeared [Frank, 2018a].

Meanwhile, the field of reversible circuit synthesis has evolved as an enabler path to quantum computing. The basic primitives proposed by Toffoli and Bennett have become the first logical gates of quantum computing [Deutsch, 1989]. Since then, this field has advanced mainly because of the expectations of high performance for quantum

computers due to their inherent parallelism [Saeedi and Markov, 2013]. Yet, despite the excitement around quantum computing, it will not replace classical computing because it addresses only a few classes of computational problems. Additionally, even with recent breakthroughs, such as the quantum supremacy experiment, constructing a universal quantum computer with many qubits will be challenging. There is a long path to reach an immune or noise-tolerant implementation [Preskill, 2018; Versluis and Hagen, 2020].

Acknowledging the inevitable end of scalability of their CMOS transistors model and conventional (irreversible) logic, the last IRDS<sup>1</sup> (International Roadmap for Devices and Systems) document presents some viable paths. Among others, two ideas can be combined in a beyond CMOS scenario: Reversible Computing as an emerging device-architecture interaction protocol, and Field-Coupled devices as an alternative to conventional transistors.

Field-Coupled Nanocomputing (FCN) devices achieve computation through local field interactions between building blocks distributed in patterns. Although this type of device has been proposed for more than two decades, Landauer’s ideas about digital devices and power dissipation have been incorporated into its design [Lent et al., 1993; Tougaw and Lent, 1994; Lent and Tougaw, 1997a; Lent and Snider, 2014]. Recently, several experimental advances have emerged in the development of its electronic version and its magnetic version [Haider et al., 2009; Pitters et al., 2011; Kolmer et al., 2015; Eichwald et al., 2014; Zografos et al., 2017; Hong et al., 2018a; Huff et al., 2018; Manipatruni et al., 2019]. Moreover, recent studies show promising advances in the development of FCN designs. For instance, placement and routing algorithms, logic synthesis algorithms of majority-based logic networks, and theoretical studies of quantum to classical transitions in FCN devices [Silva et al., 2019; Formigoni et al., 2019, 2020; Torres et al., 2020; Walter et al., 2019, 2018; Testa et al., 2018a,b; Riener et al., 2019; Neutzling et al., 2017, 2018, 2019; Zhou and Blair, 2020; Blair et al., 2018; Ramsey and Blair, 2017; Blair and Lent, 2013; Cirillo et al., 2019; Ardesi et al., 2019; Ng et al., 2020].

As shown above, there are several device-level and circuit-level innovations, but there is still a gap between these innovations and the exploration of circuit-level reversibility. The last truly physically reversible FCN system date back to 2011 [Ottavi et al., 2011]. There is a majority of other works linking FCN reversibility to FCN fault tolerance circuits and none of them are physically reversible in the given conditions [Ma et al., 2008; Thapliyal and Ranganathan, 2010; Sen et al., 2014; Roohi

---

<sup>1</sup><https://irds.ieee.org/editions/2018/beyond-cmos>

et al., 2016; Singh et al., 2017; Chabi et al., 2017; Das and De, 2017; Kianpour and Sabbaghi-Nadooshan, 2017]. Finally, reversible logic synthesis is computationally more expensive than conventional synthesis and it is mostly directed to quantum systems [Saeedi and Markov, 2013; Soeken et al., 2018; Testa et al., 2018a; Wille et al., 2019; Saeed et al., 2019]. Even when targeting CMOS based reversible logic, the primitives are not the simplest and most generic for FCN classical reversible computing [Morrison and Ranganathan, 2014; Frank, 2017; Zulehner et al., 2019].

## 1.2 Thesis Statement and Goals

This work delivers strategies to reduce fundamental energy limits based on the ideas of reversible (classical) computation applied to circuits of FCN devices. Overall, this thesis addresses the general question: "Is its possible to reduce the fundamental energy limits for a FCN circuit without any synthesis change? Specifically, can we take advantage of the current state-of-the-art irreversible logic synthesis algorithms and reduce energy limits in a post-synthesis process?"

The specific objectives of this work are:

1. Review FCN's information loss quantification and its dynamics to identify reduction opportunities.
2. Conduct an in-depth analysis of energy-throughput tradeoff for FCN partially reversible pipeline to enhance it.
3. Construct a method to reduce the energy of FCN circuit without any throughput degradation.
4. Extend the tradeoff analysis on the energy reduction of FCN circuits and other relevant metrics of digital circuits.

## 1.3 Contributions

This thesis fulfills a gap between the physics of FCN devices and efficient computation.

We present, in a novel way, physical principles related to computing and Field-coupled nanocomputing devices specifics properties. We explain Landauer's Principle, Bennett's Reversible Computing proposal, Field-coupled nanocomputing devices fundamentals, and their clocking systems.

We dramatically extend FCN partially reversible pipeline limits. First, we designed a new simulation mechanism for partially reversible pipelines and embedded it in the primary CAD tool of local field coupling devices. Next, we studied the Energy-Throughput configuration space for these systems. Both contributions were published in the Journal of Computational Electronics [Chaves et al., 2018a]. Then, we proposed algorithms capable of producing the best (optimal) configurations for a specific netlist. Precisely, given a netlist, our algorithm returns the best pipeline division based on a throughput or energy restriction. This study was published in the IEEE Design and Test journal [Ribeiro et al., 2020]. A preliminary version of this work, a heuristic version, was firstly published in the Symposium on Integrated Circuits and Systems Design [Ribeiro et al., 2018a].

We proposed and validated a new type of FCN partially reversible system based on layout manipulation and without any circuit throughput degradation. First, we created new partially reversible gates. Then, we designed post-synthesis algorithms that locally change a netlist by replacing conventional gates with our proposed new gates. Next, we validated our techniques and evaluated the results when applied to the state-of-the-art benchmarks. This study was published in the IEEE Transactions on Nanotechnology [Chaves et al., 2019]. A preliminary version of this work was firstly published in the IEEE International Symposium on Circuits and Systems [Chaves et al., 2018b].

Finally, we introduce a unified analysis with both partially reversible systems. Notably, we analyze these systems in an Energy-Throughput-Depth space, consolidating the relevant metrics influenced by our techniques. This analysis reveals the typical applicability for each system.

## 1.4 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 presents the foundations for the development of this work, i.e. the Landauer's Limit, the reversible computation proposal and the basic principles of local field coupling devices. Chapters 3 and 4 present the main contributions of this work, namely the energy improvements via clock interventions and the improvements via layout interventions, respectively. Chapter 5 presents a unified analysis of both partially reversible systems. Finally, the last chapter presents some final remarks and possible future work.

# Chapter 2

## Background

This chapter presents and explains the background for this thesis. Far from being an exhaustive review, the intention is to explicitly highlight and clarify only the fundamental concepts in the field. Section 2.1 describes Landauer’s work and briefly discusses its implications [Landauer, 1961]. Bennett’s Reversible Computing is presented in Section 2.2 [Bennett, 1973]. Next, the connection between gate abstractions and reversibility is explained in Section 2.3 [Anderson, 2013]. Finally, Sections 2.4 and 2.5 introduce Field-Coupled Nanocomputing devices and their clocking systems.

### 2.1 Irreversibility and Heat Generation in the Computing Process

In his groundbreaking work, Landauer started an analysis from basic principles. First, that digital devices are designed in such a way that the transitions between their physical states or groups of physical states necessarily correspond to the input/output relations specified by a truth table. Hence, a binary device must have at least one degree of freedom representing information. The state of a binary device is classically associated with  $k_B T$  Joules of thermal energy, where  $k_B$  is the Boltzmann constant and  $T$  the temperature of the environment. Also, to operate in a reliable way, the device must present a potential barrier between its two states with a value higher than thermal energy. Therefore, its final state is determined only by the influence of its inputs and not by noise. Figure 2.1 shows an example of a bistable potential well of a binary device.

This energy landscape represents the dynamic of a device. The device state, represented by the red dot, moves because of a cyclical energy exchange with its sur-

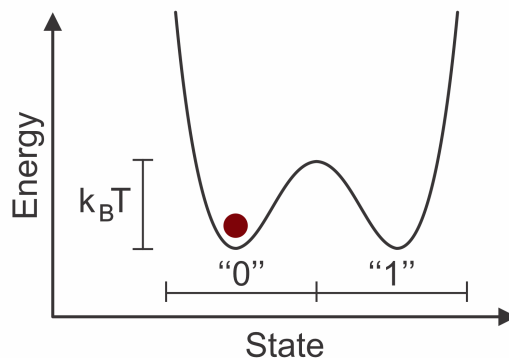


Figure 2.1: Potential energy landscape of bistable devices and system state (red dot). The example presents a system in "0" state and a barrier of at least  $k_B T$  Joules between the two states.

roundings, but it always ends in the potential energy minimum, i.e., the stable points. The external influences could come from a number of sources and types. They could be the inputs, undesired noise, or intended control signals such as clocks.

Landauer showed that the fundamental energy dissipation occurs when devices *irreversibly* erase information. He defined as logically irreversible operations those which generate outputs that do not uniquely define the operation input values, i.e., no bijective function is performed. For example, a binary device  $D$  executing a *RESTORE TO 1* operation, i.e., starting from an undefined initial state of 0 or 1 and ending in the state 1, causes irreversible erasure of information.

Landauer's argument relies on the concept of physical entropy and the implications of the Second Law of Thermodynamics. First, the usual definition of entropy in Statistical Mechanics is a measure of information,  $S = -k_B \sum_j p_j \ln(p_j)$ , where  $k_B$  is the Boltzmann constant,  $j$  labels the device data states and  $p_j$  denotes their respective probabilities of occurrence. Additionally, one should consider a device such as  $D$  (detailed above) with two equiprobable initial states and in thermal equilibrium with the environment. Then, the initial configuration has  $0.6931 k_B$  units of entropy, while the only possible final configuration has an entropy of 0. The Second Law of Thermodynamics indicates that this entropy reduction of the device must necessarily occur with compensation of greater or equal intensity in the neighborhood entropy. Thus, the energy transferred in the form of heat is at least  $0.6931 k_B T$  Joules.

As aforementioned, Landauer carried out a calculation solely based on information-bearing degrees of freedom. Therefore, as any physical computational artifact must contemplate at least these degrees, in his view, it is an unavoidable loss

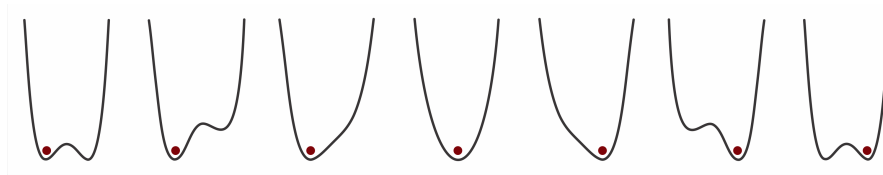


Figure 2.2: An adiabatic RESTORE TO 1 operation.

independent of the material used to implement the irreversible operation. That means fundamental energy losses result from the erasure of information, and thus, cannot be reduced by any emerging technology or process improvement. Considering its implication, it is not surprising that Landauer’s Principle has been debated since its publication.

Only after 2012, the first experiments appeared. Bérut et al. proved the principle using a system of a single colloidal particle trapped in a modulated double-well potential [Bérut et al., 2012]. Orlov et al. used a switched voltage source and an RC circuit to manage the experiment [Orlov et al., 2012]. Hong et al. performed the erasure test with nanomagnets, and, finally, Neri and López-Suárez used a micro-mechanical system [Hong et al., 2016; Neri and López-Suárez, 2016].

All these experiments share the same energetic conditions and protocol showed in Figure 2.2 and described as follows. The systems start from a double-well energy potential that is adiabatically, smoothly, distorted to reach the reset process. These tests made in entirely different physical systems confirm the generality of Landauer’s Principle.

## 2.2 Logical Reversibility of Computation

Bennett proposed a reversible Turing machine to circumvent the problem raised by Landauer [Bennett, 1973]. The idea is to design a machine that operates in three steps. In the first one, the machine performs the same steps as its irreversible equivalent would perform, but saving the intermediate results in an additional tape, ensuring the reversibility of each step. The second step consists in copying the final result to another additional tape. Finally, the original copy of the final result is used as input to reverse the computation. This third step uncomputes all signals saved in step 1. At the end of the whole process, the machine returns to its original configuration, and its tapes end only with the original input and the final result.

Bennett’s idea involves time and space tradeoffs [Bennett, 1989; Li and Vitanyi, 1996]. Reversibility can be ensured by penalizing memory, that is, by performing all



the steps that make up the desired computation and storing the intermediate results and then decomputing them. On the other hand, the whole computational process can be performed by dividing it into cycles: an intermediate step is performed, its response is copied, the step is undone, and the next one is performed, thus damaging the time in this approach.

Bennett’s proposal was considered a significant milestone because it proved that any irreversible function could be simulated by a reversible machine. This is the only possible way to perform arbitrary computations with arbitrarily low energy costs. Unfortunately, despite Bennett’s framework application example, i.e., the transcription of a DNA strand into RNA, his work lacks a demonstration of an engineered reversible device.

One of Landauer’s experimental proofs, specifically Orlov and collaborators’ experiment, also demonstrated Bennett’s idea in practice [Orlov et al., 2012]. They made the aforementioned logically irreversible operation RESTORE TO 1 mapped in a reversible system. During the erasure process, they put the system in physical contact with another artifact (which is also part of the machine) holding the same information. Thus, as the computer avoids information erasure, this change is physically reversible. In their experiment, Orlov’s group performed this reversible erasure expending only  $0.001 k_B T$  Joules. To clarify the nuances of this experiment, the following section brings a broader framework.

## 2.3 Gate Abstractions and Reversibility

Although Bennett explained his reversible protocol with an abstract machine, i.e., a reversible Turing Machine, actual computers are typically made of logic gates. Therefore, it is crucial to elucidate the logical-physical link relating gate abstractions and reversibility, which Anderson clarifies and formalizes the more general case [Anderson, 2013]. From the physical point of view, the logical transformation performed by the system is associated with the temporal evolution of a collection of interacting physical artifacts. Thus, the first step in understanding the proposed model is defining the boundaries in terms of space and time of all artifacts involved.

This is satisfied by associating a set of subsystems, shown in Figure 2.3, and a computational cycle. The  $S_{in}$  and  $S_{out}$  physical subsystems are artifacts that maintain unambiguous physical representations of input and output values through distinguishable physical states at specific moments within a cycle. The  $\mathcal{G}$  subsystem represents the physical artifact that performs the mapping between  $S_{in}$  and  $S_{out}$  states, that is,

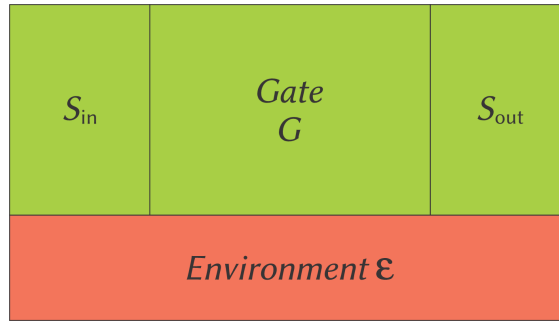


Figure 2.3: Physical gate abstraction. Adapted from Anderson [2013].

the implementation of the desired logic. The  $\mathcal{E}$  artifact represents the environment, which is the entity that receives the energy in the form of heat when the computer system ( $S_{in}\mathcal{G}S_{out}$ ) loses information.

Figure 2.4a represents the typical irreversible computational cycle with four steps performed by a logic gate that has two input wires and only one output wire. Note that  $S_{in}$  is twice the height of the  $S_{out}$  to represent that the first could carry at most 2 bits of information and the later no more than 1 bit. The first step starts with information flowing from  $S_{in}$  to  $\mathcal{G}$  (highlighted in green). Next, while information remains in  $\mathcal{G}$ ,  $S_{in}$  loses it, and  $S_{out}$  receives only a fraction. Then, in the third step, some portion of the information is irreversibly lost since it has nowhere to flow from  $\mathcal{G}$  to  $S_{out}$  (highlighted in red). The other portion is in  $S_{out}$ . Lastly, in the fourth time step, the system adjusts for a new computational cycle with  $S_{in}$  receiving new data and  $S_{out}$  losing its information.

Note that the Landauer Principle applied to this model only determines that the environment receives the information lost by the compound  $S_{in}\mathcal{G}S_{out}$  over the cycle. Yet the Principle does not determine how, at the end of the cycle, the information should be distributed among these subsystems that compose the computer. There are two options for making an operation physically reversible. The first possibility occurs when the system, even with a gate  $\mathcal{G}$  not performing a 1-to-1 mapping, has each of its components erasing information with its inputs having a copy of the same information. Take a look at Figure 2.4b. First,  $S_{in}$  transfers its information to  $\mathcal{G}$ . Second,  $S_{in}$  and  $\mathcal{G}$  hold the information while  $S_{out}$  receives a portion of the information from  $\mathcal{G}$ . In the third step, the artifact that is being fed by the computer copies the result. Then, in steps 4 to 8, the information is erased in reverse order, i.e., from outputs to inputs, always with an adjacent copy nearby. By doing so, the computer never loses information to the environment. Information always flows through computer and returns to its inputs.

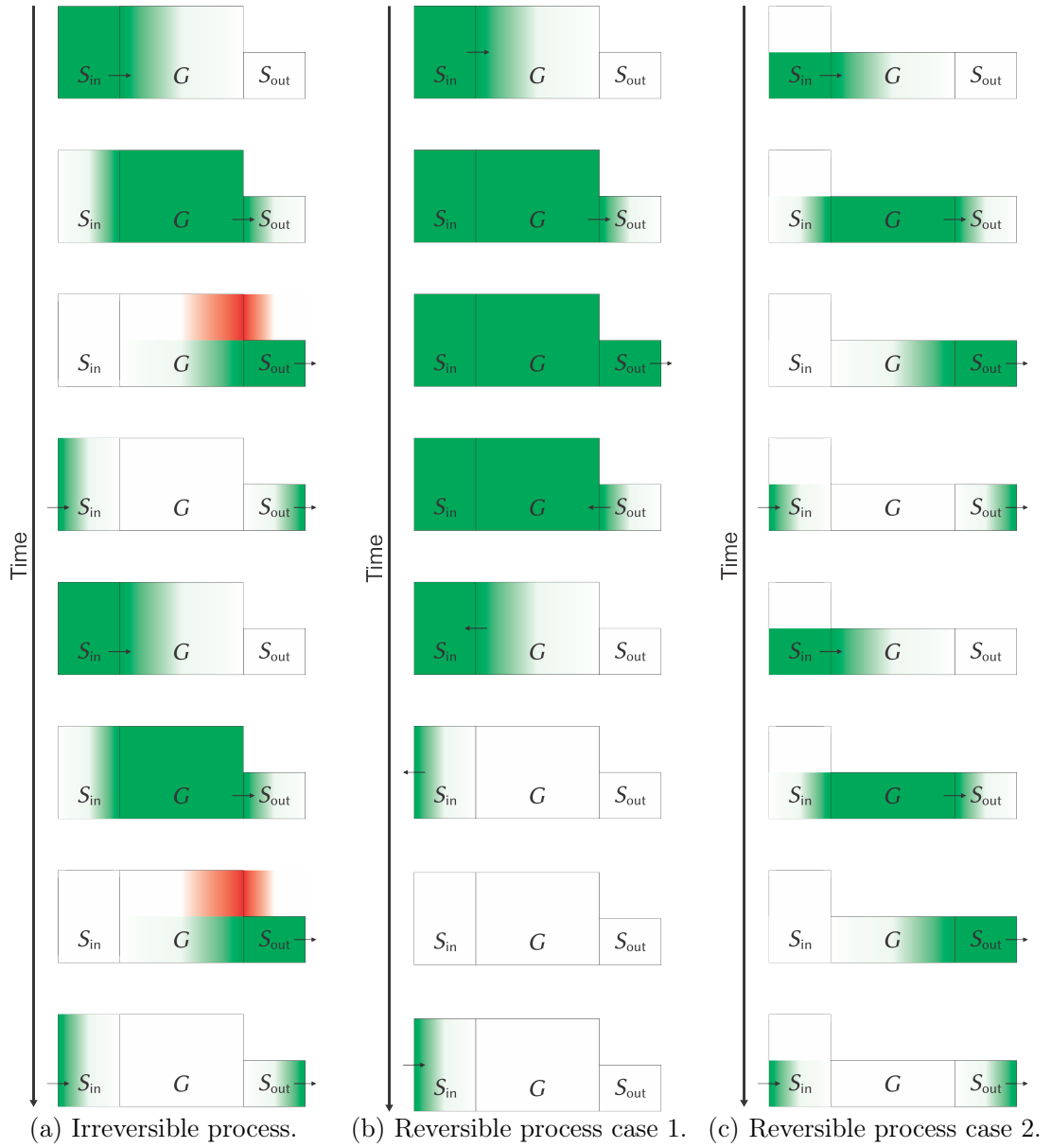


Figure 2.4: Three types of logic operations. The green color represents information flowing through a computational system. The red color shows irreversible information erasure. Irreversible process (a) presents two cycles, Reversible process case 1 (b) shows only one cycle and Reversible process case 2 (c) presents two cycles.

The other possibility occurs when the system performs a bijective relation between  $S_{in}$  and  $S_{out}$  states. In this case information and erasure flow forward from  $S_{in}$  to  $S_{out}$  through  $\mathcal{G}$ , which performs a 1-to-1 relation between  $S_{in}$  and  $S_{out}$  states. Figure 2.4c illustrates the process. First,  $S_{in}$  transfers its information to  $\mathcal{G}$ . Second,  $S_{in}$  loses its information in contact with  $\mathcal{G}$ , which holds the same information. At the same time

$S_{out}$  receives identical information from  $\mathcal{G}$ . In the third step,  $S_{in}$  holds no information,  $\mathcal{G}$  loses its knowledge while  $S_{out}$  still retains the information. Finally, step 4 completes the computational cycle, i.e.,  $S_{in}$  receives new data when  $S_{out}$  transfers the previous computation results, which contain at the end of the cycle all the information that was originally in  $S_{in}$  at the beginning of the cycle.

There is one last subtlety. After Bennett's contribution, Fredkin and Toffoli proposed special logic gates, which perform a 1-to-1 mapping [Fredkin and Toffoli, 1982]. Their reversible gates perform  $2^n$  relations linking inputs and outputs, where  $n$  is the number of input bits. Unfortunately, this requirement, a bijective connection between inputs and outputs, is more restricted than it should be. What is actually required for physical reversibility is a 1-to-1 mapping between initial and final physical states. Take, for example, some gate in a specific place in a netlist where only occurs a subset of its  $2^n$  inputs combinations with just bijective relations. This gate is logically reversible, and it could be physically designed to reach arbitrarily low energy levels [DeBenedictis et al., 2016]. Frank formalized and named this concept as "Conditional Reversibility" [Frank, 2005, 2017]. In conclusion, the system could not expend any amount of energy in the form of heat.

## 2.4 Field-Coupled Nanocomputing Circuits

Field-Coupled Nanocomputing (FCN) devices are a potential alternative to traditional CMOS technologies [Anderson and Bhanja, 2014]. Lent et al. proposed this type of digital device based on four essential ideas [Lent et al., 1993]. The first consists in using the position of the charges to encode Boolean states. This feature comes from the ability to create structures known as quantum dots that are capable of trapping electrical charges. The second idea relates to the robustness of devices. According to Landauer, devices that exhibited an bistable information transfer function with saturation would be more tolerant to variations in the manufacturing process and environmental disturbances [Landauer, 1989]. The third idea is the nonlinearity of charge tunneling between such dots because of charge quantization. Finally, the last idea is the notion of a local coupling architecture as an analogy to cellular automata.

FCN devices are based on nanoscale cells that interact via local fields in absence of any current flow. This enables tremendously low energy dissipation and motivated numerous contributions in the past. Several physical realization of FCN devices have been developed both for its electronic version, Quantum-dot Cellular Automata (QCA), and for its magnetic version, Nanomagnet Logic (NML) [Haider et al., 2009; Pitters et al.,

2011; Eichwald et al., 2014; Zografos et al., 2017; Hong et al., 2018b]. Moreover, recent studies show promising theoretical advances in the development of FCN designs [Lent et al., 2016; Walter et al., 2018; Torres et al., 2018; Ribeiro et al., 2018a,b].

In case of QCA, each cell consists of four quantum dots, i. e., structures able to confine electric charges. These quantum dots are arranged in a square-like fashion such that free mobile electrons can move between them. Since these electrons impose mutual repulsion due to Coulomb interaction, they tend to locate themselves at opposite corners of the square [Lent and Tougaw, 1997b]. Thus, they assume stable states, called polarizations, which are energetically equal and interpreted as binary 0 and 1. When placed close to each other, the polarization of one QCA cell influences the polarization of the other—again by Coulomb interaction. One can exploit this effect to construct logic gates, such as the *NOT* and the *MAJ* gates. Finally, locking one of the inputs of a *MAJ* gate to 0 or 1-state creates an *AND* or an *OR* gate, respectively.

NML circuits operate in a similar fashion but, instead of cells with quantum dots, the technology has single-domain nanomagnets as basic blocks. The magnetization of each nanomagnet is associated with binary 0 and 1, where both states are energetically equal as in QCA. Again, when placed close to each other, the magnetization of one nanomagnet influences the magnetization of its neighbors. The *MAJ* gate has the same layout as its QCA counterparts, while the *NOT* gate is even simpler through antiferromagnetic coupling [Imre et al., 2006].

To build more complex FCN devices, one does not only needs to select the placement of cells carefully but also needs to synchronize the information, avoiding a signal to reach a logic gate and propagate before the other inputs reach the gate. This is achieved by clocking systems and is crucial for FCN circuits, ensuring its correct operation.

## 2.5 Clocking Systems

The clocking system is an important factor in the dynamics of FCN circuits. Its principal functions are the synchronization of data flows and the implementation of adiabatic operation, enabling circuits with high energy efficiency. In QCA circuits, this efficiency is archived by adiabatic switching typically divided in four phases [Lent and Tougaw, 1997b; Lent et al., 2016]. The dynamics of NML circuits can be controlled by clock signals with three and four phases. However, lowest energy consumption is also only possible with the latter [Lambson et al., 2011; Martini et al., 2016].

The clock is an electrical field which controls the tunneling barriers within a cell,

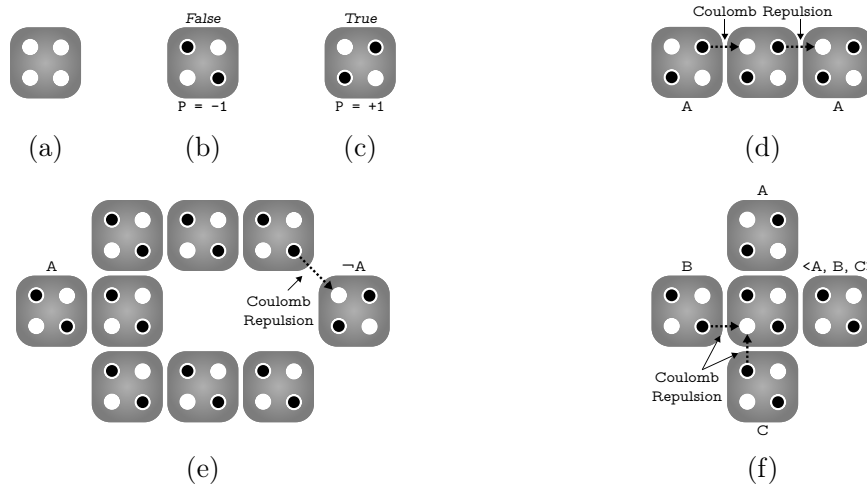


Figure 2.5: QCA: (a) A basic cell. (b) A cell fixed on *false*. (c) A cell fixed on *true*. (d) A wire. (e) An inverter gate. (f) A majority gate.

thus keeping control when a cell might or might not be polarized [Lent and Tougaw, 1997a]. It can be applied to groups of cells (clock zones). In each zone, a single potential can modulate the barriers between the dots. The scheme of clock zones permits a cluster of QCA cells to change their polarization accordingly to its neighbors from the previous zone, performing a certain calculation. It also allows the same cluster to have its states frozen, acting this way as inputs to the next clock zone.

QCA's clocking phases, named *Switch*, *Hold*, *Release*, and *Relax*, do this feature. On the *Switch* phase, the cell starts with its tunneling barriers low, growing gradually, allowing the cell to polarize according to the state of their input cells. On the next phase, *Hold*, its barriers stay high while the cell holds its polarization and can be used as input to the next stage. On the *Release*, the barriers lower gradually, allowing the cell to depolarize. Finally, on the *Relax* phase, the barriers continue lowered while the cell stays relaxed.

Landauer clocking scheme has four different phases [Lent and Tougaw, 1997a]. Figure 2.6 shows an example. In the first time step, the cells in clock zone 1 are in *Switch*, which means that they start depolarized with low tunneling potential barriers. During this phase, the barriers between the dots are progressively increased, and the cells begin to polarize according to the state of their drivers (their input or neighbor cells). At the end of the first step, the barriers are high enough to avoid the tunneling of any electron, so the cells' states are fixed.

During the second time step, clock zone 1 is in *Hold*, the cells have fixed states as the barriers are kept high. So they can be used as inputs to the cells in clock zone

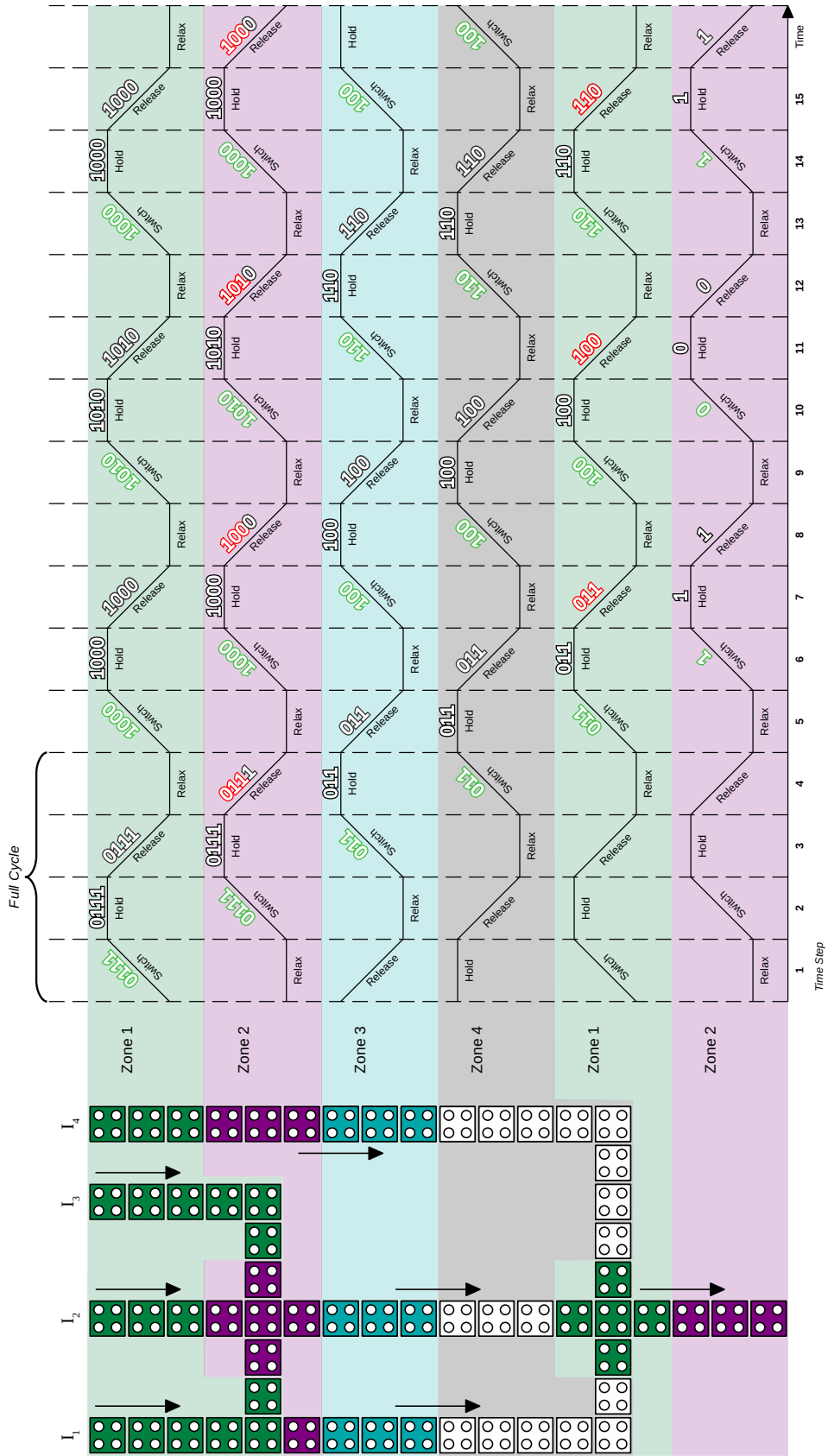


Figure 2.6: An example of Landauer clocking scheme controlling the polarization of cells in a circuit with two majority gates. Each curve represents the tunneling potential barriers of its respective zone. The circuit loses information in each clock cycle in the zones with majority gates (highlighted in red).

2, which are in *Switch*.

In the third time step, the cells in clock zone 1 enter the *Release* phase, where the barriers are progressively lowered, and the cells are allowed to relax to a depolarized state. The cells of clock zone 2 (*Hold*) polarize the cells in clock zone 3 (*Switch*). One can see that, until this step, all input information is still preserved in the circuit. However, it is not the case for the fourth time step.

In the fourth, zone 2 starts to release its state and then irreversibly erases part of circuit's inputs highlighted in red. Despite this issue, Landauer's clock has a fixed throughput that is independent from the depth of the circuit due to its inherent pipeline. We have new output values for each 4 time steps, a full cycle. Therefore, any circuit with Landauer's clock has a fixed throughput of  $1/4$  Operations Per Time Unit (OPTU).

An alternative to avoid this irreversible erasure of information is to change the clocking signals in a way that the circuit can naturally return to its initial state after the computation. This requires a modification in the timing of the clocking signals. This way the information is simply held in place by the clock until a computational block is complete, then erased in the reverse order of computation [Lent et al., 2006]. This is called Bennett clocking. Figure 2.7 shows an example.

In the first time step, the cells in clock zone 1 are in *Switch*, meaning that they can polarize according to the polarization of the input cell. At the same time, the cells in clock zones 2 through 6 are in *Relax*.

In the second time step, clock zone 1 is in *Hold*, clock zone 2 changes to *Switch* and clock zones 3 through 6 remain in *Relax*. This pattern lasts until clock zone 6 reaches *Hold*, as one can see in the seventh time step.

Then, in the eighth time step, clock zone 6 changes to *Release*, while the other clock zones are in *Hold*. In the ninth time step, clock zone 6 is in *Relax*, clock zone 5 is in *Release* and all other clock zones still in *Hold*. Again, this pattern lasts until all clock zones reach *Relax* phase. One can see that, in all steps, the information is reversibly erased since a copy still exists in an adjacent cell.

Despite this six zones example, the Bennett clocking strategy works for an arbitrary number of zones. Nevertheless, this scheme induces a delay in signal propagation, since all cells in a circuit must be totally polarized, and then completely depolarized in reverse. This disrupts the pipeline of QCA circuits when operating with Landauer clock. Note that we use the same layout in Figures 2.6 and 2.7. This circuit needs 4 time steps to transfer information under Landauer clock scheme. Therefore it has a throughput of  $1/4$  OPTU. Under Bennett's clocking scheme, the same circuit needs 14 time steps to transfer information. Therefore, it has a throughput of  $1/14$  OPTU. We



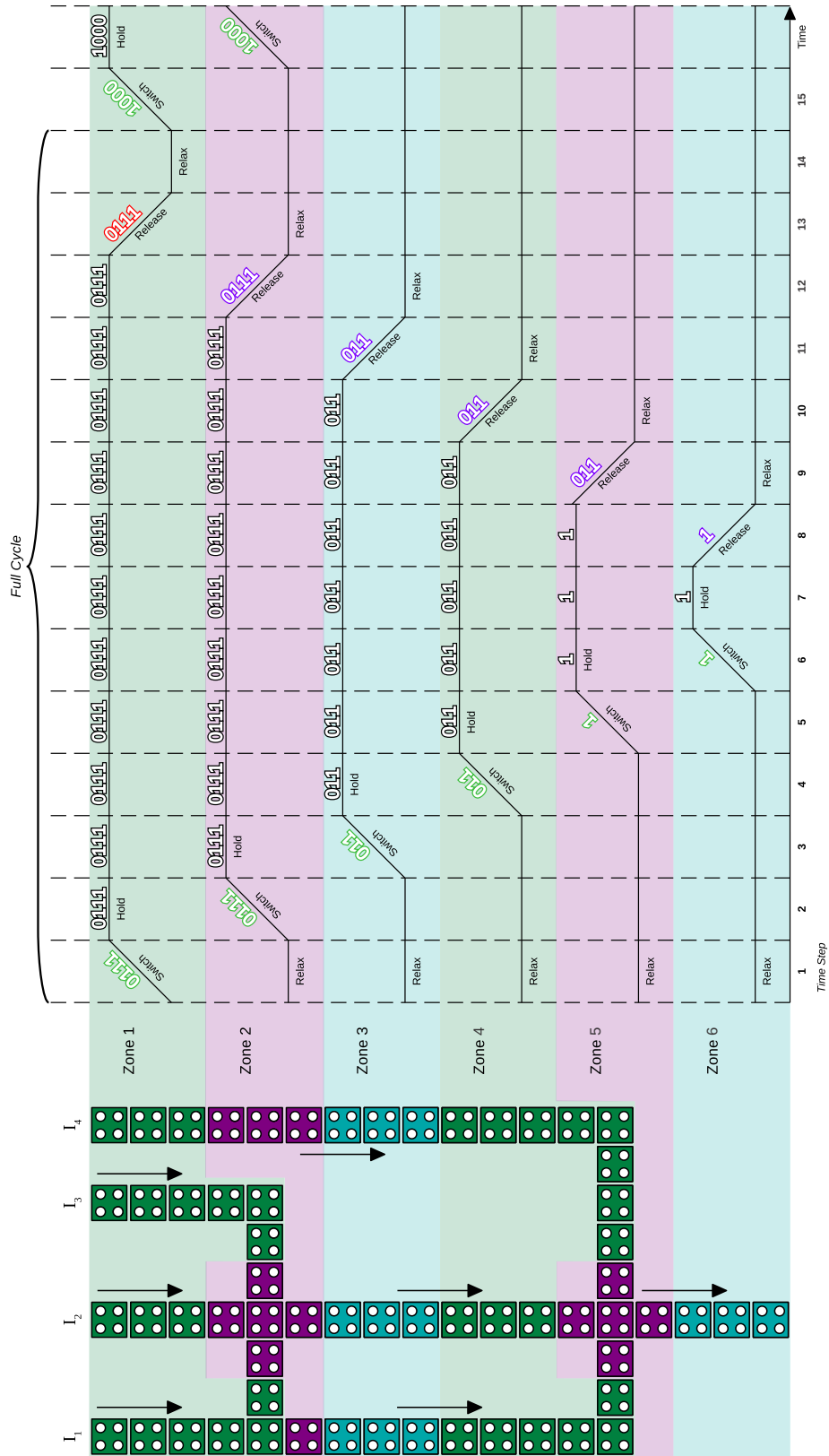


Figure 2.7: An example of Bennett clocking scheme controlling the polarization of cells in a circuit with two MAJ gates. There are no information losses within the circuit since each erasure happens with an adjacent copy (highlighted in purple).

can generalize the cycle size under Bennett's operation:  $2n + 2$ , where  $n$  is the number of zones. Thus, the throughput in all Bennett's schemes is  $1/2n+2$  OPTU.

To minimize the damage in throughput and preserve some gain in energy consumption, Ottavi et al. [2011] proposed an architecture which divides a circuit into smaller Bennett clocking regions or stages. Instead of waiting for the full circuit to switch and then relax, the system could recover its pipeline attribute due to its division in stages. As soon as any stage's computation result reaches its output, the next stage starts to switch, and the current stage begins to release.

The Figure 2.8 shows an example of a circuit with two stages (Bennett clocking regions), each of which has three Bennett's zones. First, the zone 1 in stage 1 switches while the other zones of the same stage are in relax phase. In the second time step the aforementioned zone 1 stays in hold phase while the following zone switches. This pattern continues until the information reaches the current stage's output and the next stage's input as we could see in steps 1 to 4. In the fifth time step the different dynamics from the previous case (Bennett clocking) arises: the last zone of the first stage starts to release while the second zone of the second stage starts to switch. So, while the data moves through the second stage to polarize its output, the first stage lose its polarization related to the previous computation as we can see in steps 5 to 8. After that, the system ends its cycle returning to its original state.

Note that Bennett's clocking scheme is a particular case of Ottavi's pipeline when the number of stages is equal to 1. For this reason the throughput is also  $1/2n+2$  OPTU, where  $n$  is the number of zones in a stage. One sees that losses always occur once per cycle between stages (cycles 7 and 10 in Figure 2.8).

The three configurations, *i.e.*, Landauer's, Bennett's and partially reversible pipelines, have differences regarding information/energy losses. In Landauer's case, the losses are a property of the layout. They are a result of locally states' merge, occurring within the conventional logic gates, *e.g.*, AND, OR, MAJ. In these gates,  $n$ -bits inputs are merged in some particular way to produce 1-bit output. It is possible to build reversible systems under this clocking scheme if the designer uses reversible logic gates (with a 1-to-1 relation between gates' initial and final states) instead of the conventional ones [Lent et al., 2006]. The other typical artifacts in logic netlists (inverters, wires, and fanouts) are already reversible ones. This strategy will be explored in Chapter 4.

The dynamics of the Bennett's clocking are reversible independent of the layout. Figure 2.7 shows that the circuit can always return to any previous state exactly in reverse order if the zone's control signals are time-reversed. At the end of a cycle, the losses within the circuit are negligible, and only its primary inputs could be lost [Lent

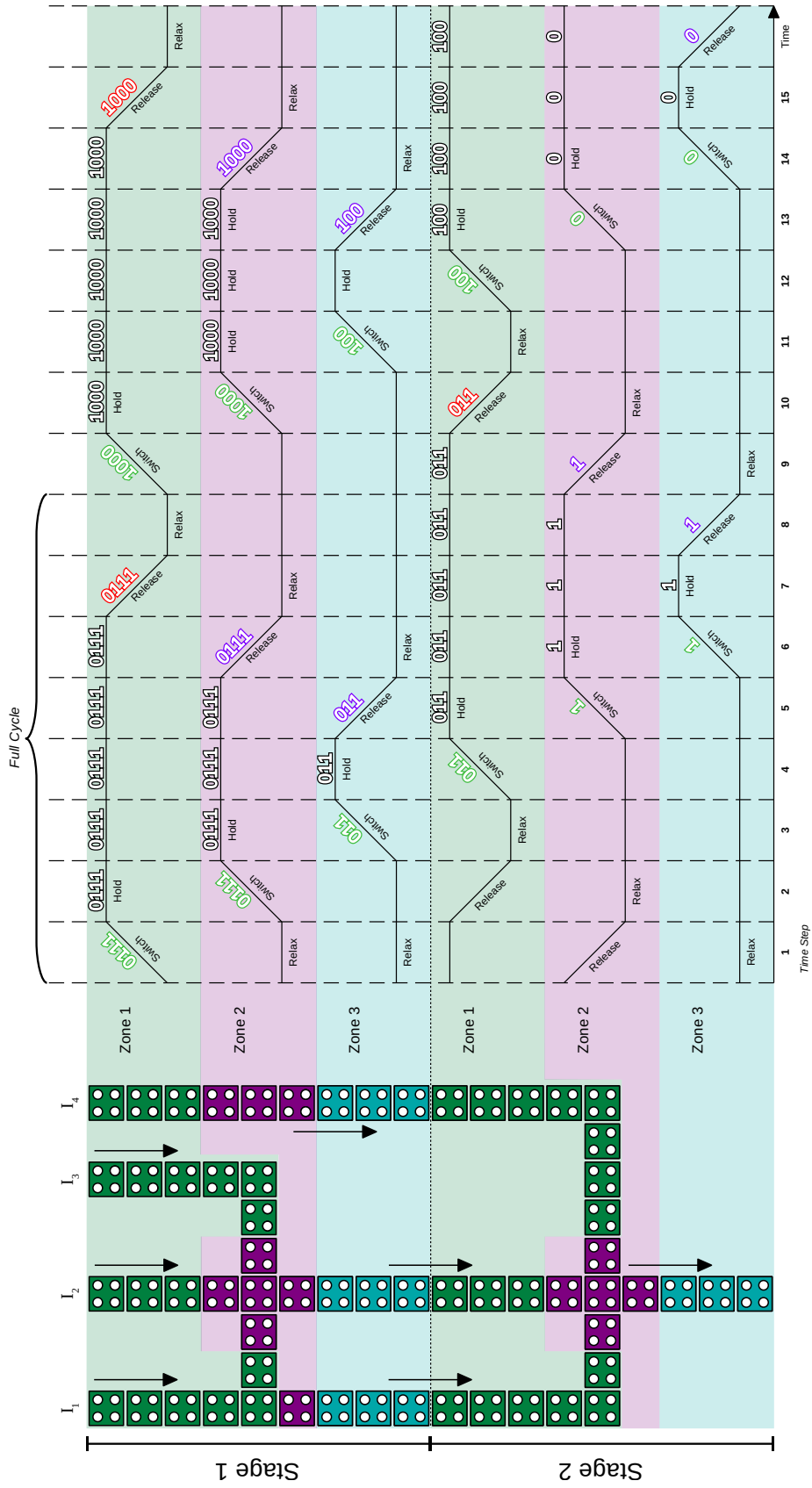


Figure 2.8: An example of a partially reversible pipeline with two stages, each of which with three Bennett zones. The dotted line represents the division between stages. Stage 2 loses information each clock cycle in its input's cells since there are no adjacent copies (highlighted in red).

et al., 2006].

The partially reversible pipeline is an intermediary case since each stage is exactly a Bennett clocked circuit. Consequently, the losses take place in each stage's inputs. Note that this means that even wires that connect stages are no longer reversible. This strategy will be explored in Chapter 3.

Let's take the layout in Figure 2.8 as an example. In Landauer's clock, the losses will occur in the two MAJ gates. In Bennett's clock, they will only occur in the circuit's primary input. Finally, in the partially reversible pipeline, it will occur in the two stages inputs.

## 2.6 Summary

Computers do operations over random data. These machines typically do not know about their input probabilities; they only do deterministic transformations on their input data. This knowledge of input probability distribution, quantified by physical entropy, is linked to energy fundamental limit and heat generation. As mentioned previously, heat could be eliminated through mapping computation in a reversible process. Reversibility is related not only to the device's mapping but also to its boundary conditions. All information lost by a logic gate must flow to another part of the computer, either inputs or outputs, so it is not thermalized.

Field-Coupled Nanocomputing devices were conceived from Landauer's ideas. Hence, their energetic losses are mainly related to how clocking and layout together allow information erasure. Our contributions to reduce such losses are presented in the next two chapters. Chapter 3 presents techniques based on timing manipulation. Afterwards, Chapter 4 presents a new proposal for a partially reversible system based on layout manipulation.

# Chapter 3

## Partial Reversibility via Clocking

The notion of Partially Reversible QCA Systems was first employed by Ottavi et al. [2011]. Here, we propose new contributions to enhance this structure. The first contribution is an algorithm enabling the simulation of clock-based reversible systems in the main QCA CAD tool, QCADesigner [Walus et al., 2003]. Another significant contribution is our exploration of the energy/throughput tradeoff in the state-of-the-art benchmark suite. This analysis allows us to identify valuable opportunities to improve both metrics. Finally, we propose an optimal algorithm to return the best energy dissipation given a throughput restriction and the opposite, i.e., the best throughput given an energy dissipation restriction.

### 3.1 Partially Reversible Pipelines in QCADesigner

QCADesigner is the main tool used to simulate QCA circuits [Walus et al., 2003]. This software performs simulations based on two engines. The first and simpler type, called bistable, assumes that each QCA cell is a simple two-state system and does not take into account any timing consideration related to the quantum-mechanical evolution of the system. The second type, namely coherence vector engine, is based on the density matrix approach, which models the power dissipative effects of QCA cells. Unlike the bistable engine, the latter performs a time-dependent simulation of the QCA circuits. Although one of the engines presents energy considerations, its calculations are not information oriented. It does not consider losses related to information erasure. It is worth to mention that QCADesigner uses the tunneling probability as clock signals rather than the tunneling potential barriers.

The latest version of the QCADesigner only performs Landauer clocked circuit simulations. In this case, the pipelined dynamics in Figure 2.6 results from the use of

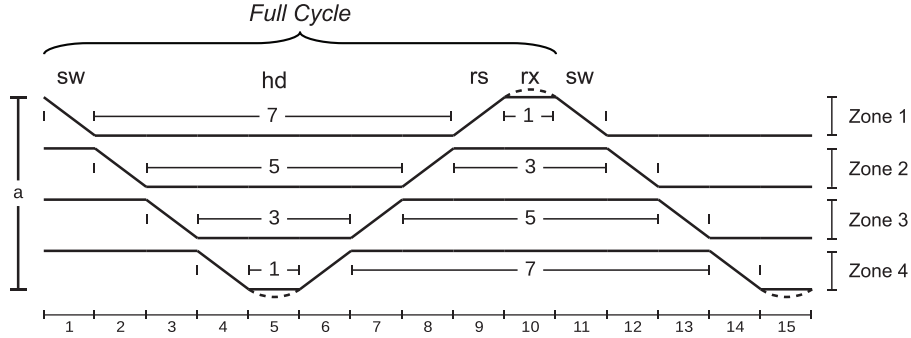


Figure 3.1: Wave base for Bennett's clocking signals generation. QCADesigner uses the tunneling probability as clock signals rather than the tunneling potential barriers. The clocking phases *Switch*, *Hold*, *Release* and *Relax* are indicated by sw, hd, rs and rx, respectively.

four clocking waves shifted by  $\pi/2$  as potential barrier modulators. QCADesigner calculates the clock signal as a hard-saturating cosine and ties it directly to the tunneling energy. To implement this clocking scheme in the software, they pre-established and used four clocking waves to switch the circuit accordingly. Those waves are calculated as being truncated cosine in a way that *Hold* and *Relax* phases have the same duration.

In the Bennett clocking scheme, *Hold* and *Relax* phases have different, but specific relations. In a circuit with four Bennett clock zones, as in Figure 2.7, the first clock zone stays 7 time steps in *Hold* and 1 time step in *Relax*. The second zone, in its turn, spends 5 steps in *Hold* and 3 in *Relax*. The third zone stays 3 steps in *Hold* and 5 in *Relax*, and, finally, the last zone stays 1 in *Hold* and 7 in *Relax*. Equations (3.1a) and (3.1b) show a way to generalize those relations for an arbitrary number of zones:

$$r = 2i + 1 \quad , \quad i \in [0, k) \quad (3.1a)$$

$$h = 2k - r \quad (3.1b)$$

where  $i$  is the  $i$ -th clock zone,  $k$  is the total number of zones,  $r$  and  $h$  are the numbers of time steps in *Relax* and *Hold* phases, respectively.

As the ratios between *Hold* and *Relax* are already known, we use it to generate Bennett's clocking signals. Using  $a \times \cos(\omega t) + b$  as a base function, the task is to slice this function to generate waves as those in Figure 3.1. Here,  $\omega$  and  $t$  are the clocking frequency and time, respectively. The parameter  $a$  is a multiplicative factor of the cosine function. It is used to enhance the function curve, decreasing or increasing the gradient of the zones in *Switch* and *Release* phases. The parameter  $b$  is the offset of

---

**Algorithm 1:** Bennett clock signals generation.

---

```

data : A: clock amplitude
data :  $\omega$ : clock frequency
result : nested waves related to Bennett clocking zones
1  $cdata \leftarrow \emptyset$ 
2 foreach zone in clocking zones do
3    $cdata[zone] \leftarrow \emptyset$ 
4   for  $t \leftarrow t_0$  to  $t_f$  do
5      $val \leftarrow \min(1, A \times \cos(\omega t) + b_{zone})$ 
6      $val \leftarrow \max(-1, val)$ 
7      $med \leftarrow (cmax - cmin) / 2$ 
8      $cdata[zone][t] \leftarrow val \times med + cmin$ 
9 return  $cdata$ 

```

---

the cosine function between the clock zones.

We carried out a numerical analysis in order to find the parameters that lead to a relation between *Hold* and *Relax* that met the required ratios previously established. Then the value of clock for each time step in the simulation can be calculated. The clocking algorithm of the QCADesigner was modified, as seen in Algorithm 1. The outer loop at line 2 iterates over each clock zone. The inner loop at line 4 has its iterations representing time steps. Lines 5 and 6 implements clocking signal saturations, i.e., they ensure that the clock is always between -1 and 1. Finally, lines 7 and 8 shift the waves vertically so that the values are within the expected range.

It is simple to extend this implementation to allow partially reversible pipelines simulations. For a circuit with  $n$  zones to be divided in  $m$  stages, the zone control signals of each stage differs only by a  $\pi n/m$  shift, i.e., the first clocking wave of the  $n$  stage has the shape of the first clocking wave of the  $n + 1$  stage but delayed as can be seen in Figure 3.2.

We performed three computational experiments to validate our algorithm using the coherence vector engine of the modified QCADesigner 2.0.3.

First, we simulated the transmission of data through a wire (Figure 3.3). The circuit presents one stage with four Bennett clock zones (which imply in a cycle of 10 time steps wide), using 21 cells in a  $1 \times 21$  cells area. We measured the polarization of the last cell in each zone and it was as expected, i.e., each one kept its polarization only up to the moment when the polarization and depolarization of remaining zones happened.

Second, to test a bigger structure, we simulated the operation of a 4-to-1 multiplexer (Figure 3.4a). The circuit presented two stages with four Bennett clock zones

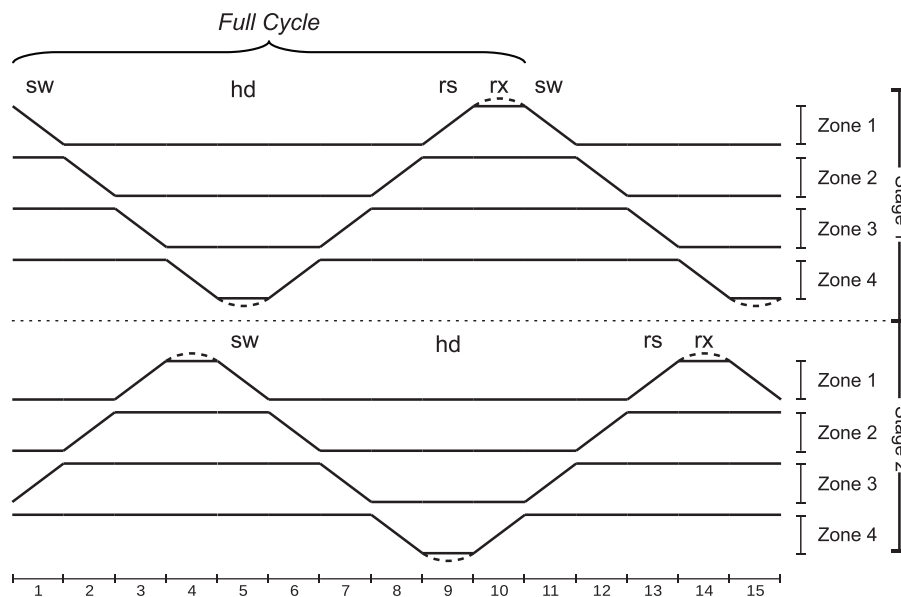


Figure 3.2: Clocking signals for a circuit with 8 zones divided in 2 stages.

each (which imply in a cycle of 10 time steps wide), using 184 cells in a  $35 \times 50$  cells area. Despite testing the multiplexer with all possible inputs combinations, Figure 3.4b only shows the simulation results of a few cases for clarity.

Finally, to test a structure with more Bennett clock zones, we simulated the operation of a 2-bit adder (Figure 3.4c). The circuit presented two stages with 16 Bennett clock zones each (which imply in a cycle of 34 time steps), using 997 cells in a  $99 \times 80$  cells area. Figure 3.4d shows that it works properly for all possible input combination.

One can see that the clocking signals fulfill their role as cell potential barrier modulators. They allow the cells to polarize only when their clocking signal is at its lowest level (QCADesigner actually displays the tunneling probability as clock signals rather than the tunneling potential barriers).

All presented designs also work under Landauer clock without any change in the cells' placement. The change in the number of clock zones in partially reversible pipelines allows a balance between performance and energy efficiency, as demonstrated in Ottavi's work [Ottavi et al., 2011]. Our presented adder could be faster if we divide our design into more stages and fewer Bennett zones per stage. However, the cost will be the energy consumption increase. We will explore this tradeoff in the following section.



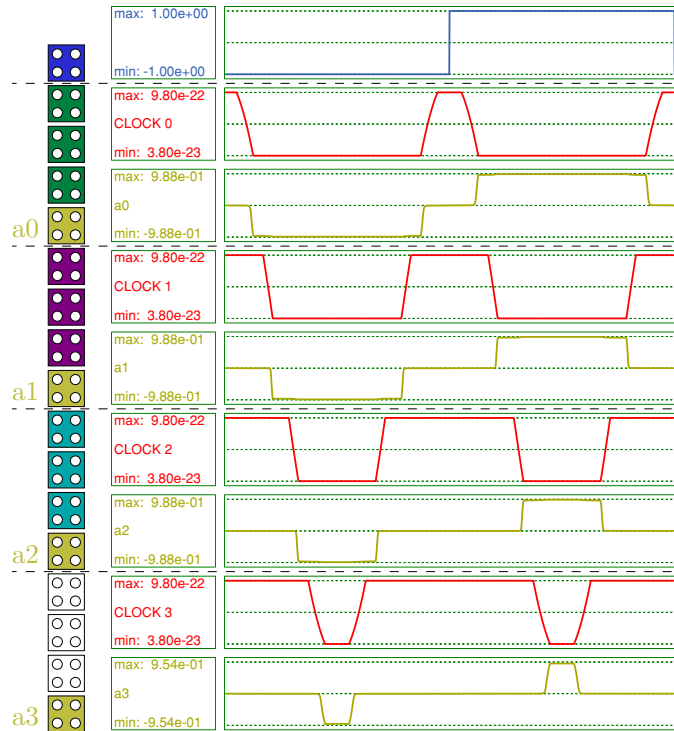


Figure 3.3: Validation of our Bennett clocking implementation.

## 3.2 Energy-Throughput Relations

The clock scheme impacts directly in the following circuit's performance metrics: throughput and energy. The first metric is obtained from the clock size (Section 2.5). Therefore, for Landauer's configuration, it has a fixed value of 0.25 OPTU due to its inherent pipeline. In Bennett's case and its variations, the throughput depends on how we divide the layout into stages and is evaluated as  $1/2^{n+2}$  OPTU, where  $n$  is the number of zones inside a stage.

We evaluated the energy dissipation using a method similar to Ottavi's [Ottavi et al., 2011]. Despite that his implementation of the Parity Checker and Full Adder was at a lower level with AND, OR and MAJ gates, he evaluated at a higher level considering XOR and 1-bit Full Adders as monolithic blocks, counting the number of input bits for each block in the design as energy losses. In our work, for Landauer's clock scheme we count the number of input bits for each lower level gate in the design as energy losses. For Bennett's and its variations, we followed exactly Ottavi's evaluation, considering the number of input bits in the stages as losses.

To evaluate the impact in performance for different clock configurations, we used the EPFL Combinational Benchmark suite [Amarù et al., 2015]. These benchmarks present 20 circuits that are divided into classes according to the nature of their function

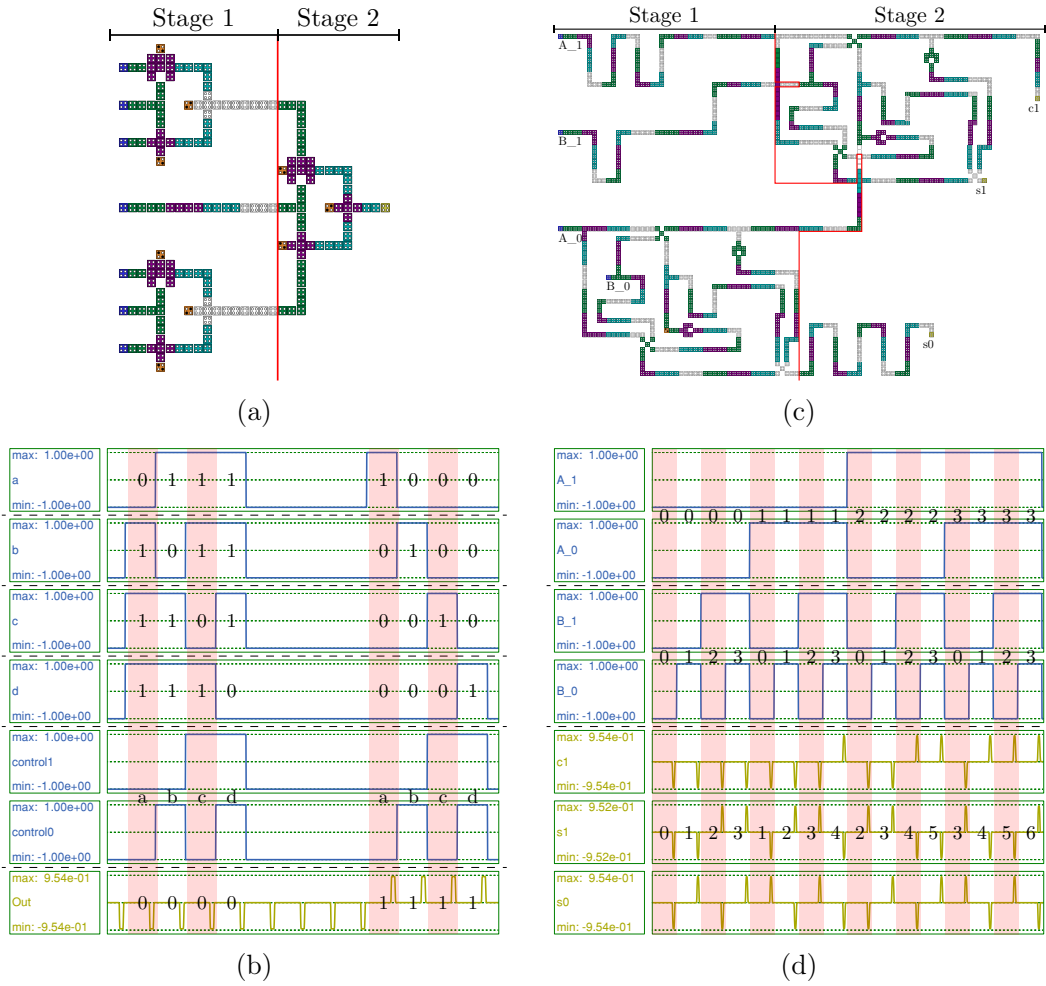


Figure 3.4: 4-to-1 multiplexer: (a) circuit layout and (b) functional simulation results. 2-bit adder: (c) circuit layout and (d) functional simulation results.

being 10 arithmetic circuits and 10 related to the random/control functions. We used the benchmark’s logic netlists exactly as it was released. We consider the number of gates on the circuits’ critical paths as the number of zones between circuit’s primary inputs and outputs. Therefore, we are considering the best (minimum) delays.

Figures 3.5, 3.6 and 3.7 show features of various configurations of three different circuits, Adder, Memory Controller and Sine. These specific circuits were chosen because they have different loss patterns, but similarities to the other seventeen benchmark circuits. Thus, these three represent well the whole set according to the relevant characteristics analyzed here, energy and throughput.

First, Figures 3.5a, 3.6a and 3.7a presents in the y-axis the accumulated energy loss, and in the x-axis presents the layout’s zone where this loss occurs. Each zone corresponds to a level. According to Section 2.5, the losses in Landauer’s clock occurs

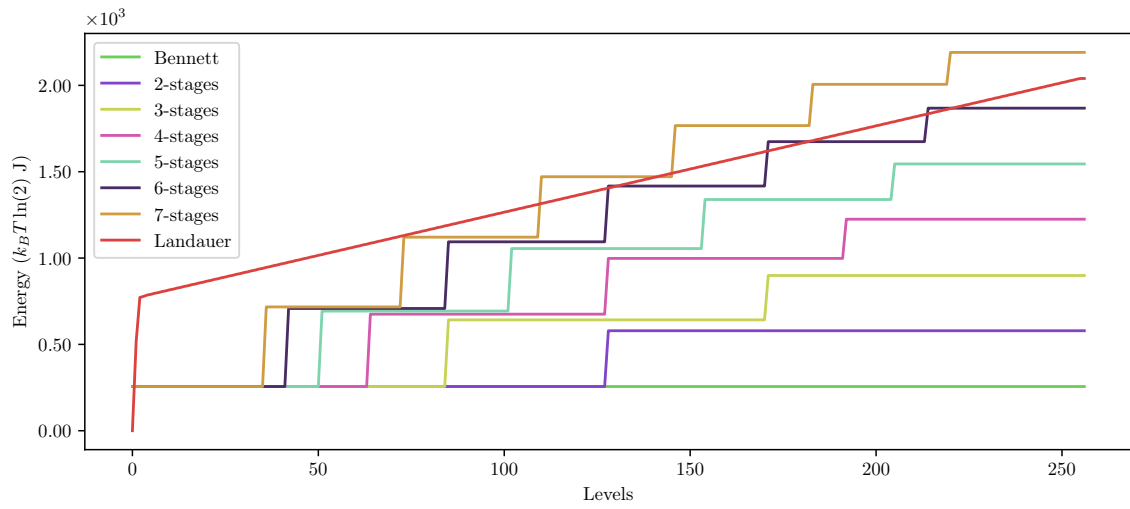
at each logic gate. On the other hand, in Bennett's configurations, the losses take place in the stage's input. Therefore the number of energy steps in the Bennett's curves is equal to the number of stages.

Figures 3.5b, 3.6b and 3.7b show the number of wires and the number of gates' input bits in each level of the circuit. Figures 3.5c, 3.6c and 3.7c show the configuration space of each circuit. These images show the Energy and Throughput values for the Landauer, Bennett and Partially Reversible Pipeline configurations. The latter is divided into non-dominated solutions, which configurations are better than the others in at least one aspect (energy or Throughput), and the dominated solutions, which are worse than the others in all aspects under evaluation.

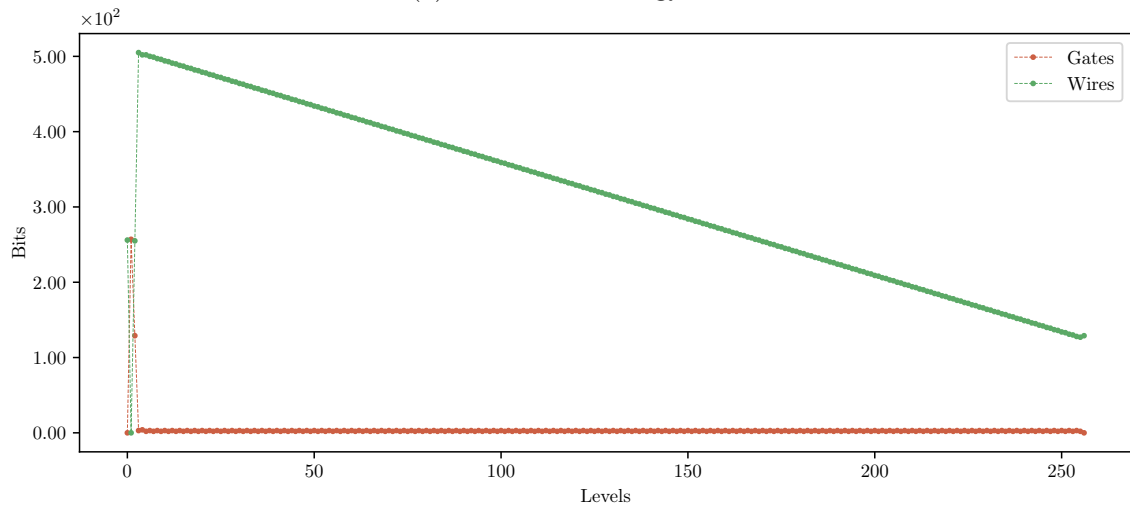
The first group contains circuits whose losses are almost constant through its data path. It encloses the Adder, the Divisor, the Square-Root and the Round-Robin Arbiter. Figure 3.5a shows the Adder's accumulated energy loss for different clocking configurations. One can see that for Landauer's case, despite the layout's first zones losses, we have a linear pattern. Even in Bennett's configurations, the pattern lasts, *i.e.*, the step height stays almost constant. This pattern can be explained by analyzing Figure 3.5b, which shows the number of wires and gate's input bits in each circuit level. As the 7-stages partially reversible pipeline is energetically worse than Landauer's configuration, dividing this circuit into 7 stages or more is not worth. Figure 3.5c shows the entire configuration space for Adder's circuits. Bennett's configuration presents the best energy (smallest) and the worst throughput (smallest). Landauer's configuration, on the other hand, presents the best throughput (highest). Therefore, as the number of stages rises, the throughput improves, and the energy deteriorates. We can conclude that these divisions are only useful while the configurations have at least one metric better than Landauer's, energy or throughput.

The second group contains circuits whose losses decay as levels become deeper. It encloses the Memory Controller, the Square, the Coding-Cavlc, the i2c Controller, the Int to Float Converter, the Priority Encoder, the Lookahead XY Router and the Voter. Figure 3.6a shows the accumulated energy loss for the Memory Controller. There is a common pattern behind Landauer's and Bennett's configurations: the loss rate diminishes as information approaches the outputs. The reason is the high decrease in wires density from the circuit's primary inputs to outputs, as Figure 3.6b shows. Figure 3.6c shows viable configurations for Energy and Throughput, except for the highest gray point (11-stages partially reversible pipeline).

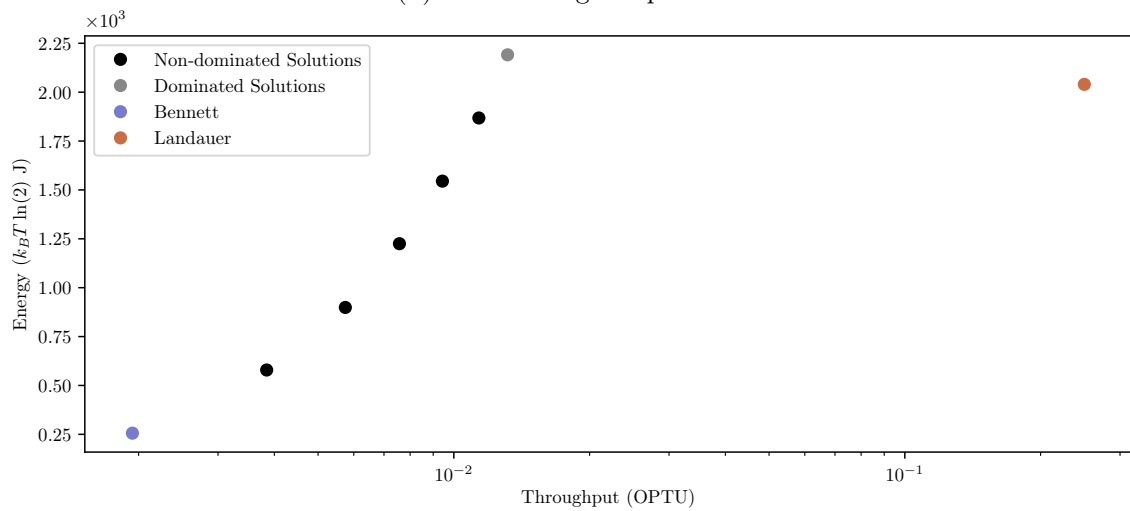
The third group contains more complex circuits regarding losses rate. It encloses the Barrel shifter, the Hypotenuse, the Log2, the Max, the Multiplier, the Sine, the Alu Control unit, and the Decoder. Figure 3.7a shows the Sine circuit. It has different



(a) Cumulative energy loss

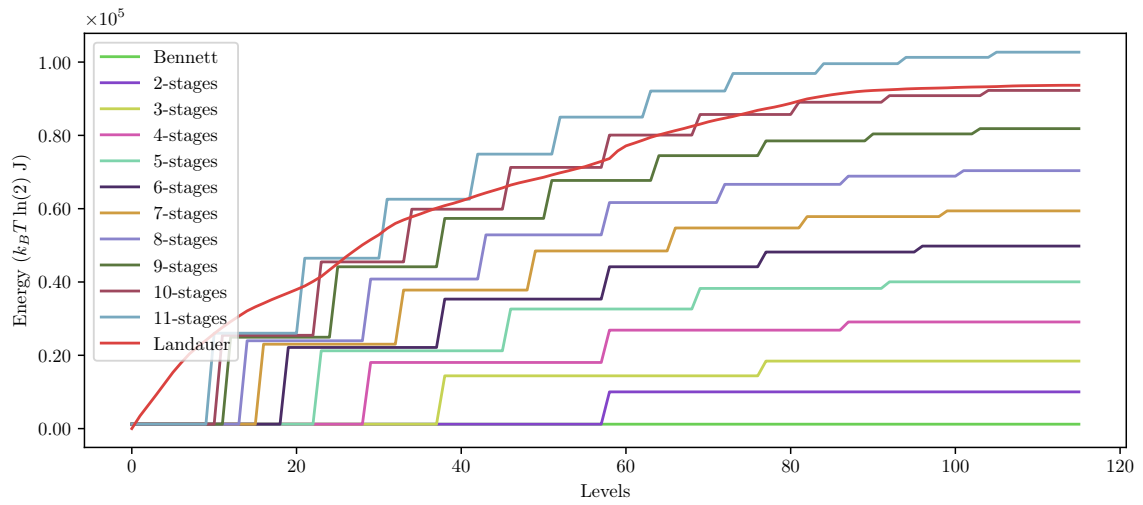


(b) Wires and gates per level

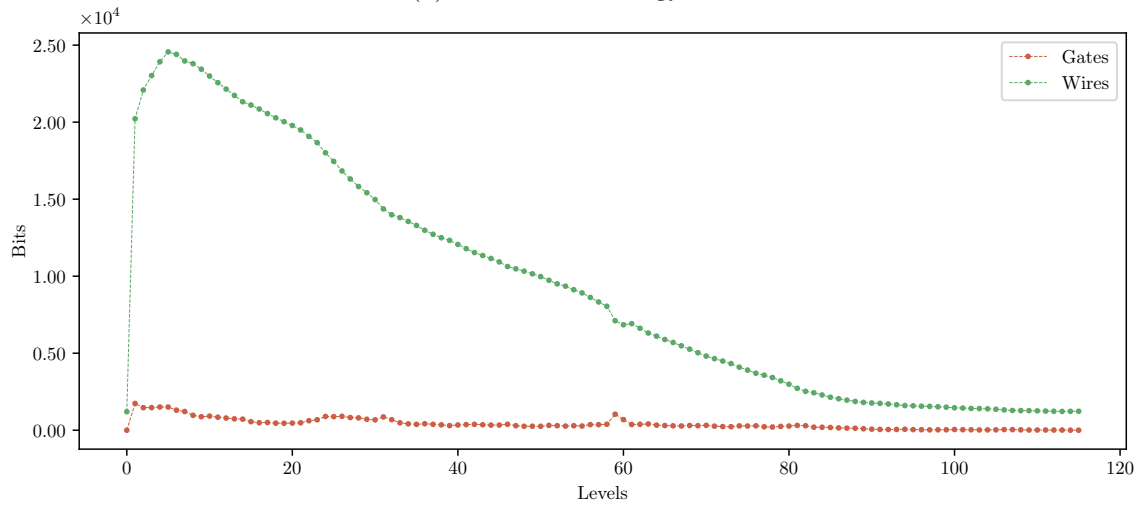


(c) Energy *versus* Throughput

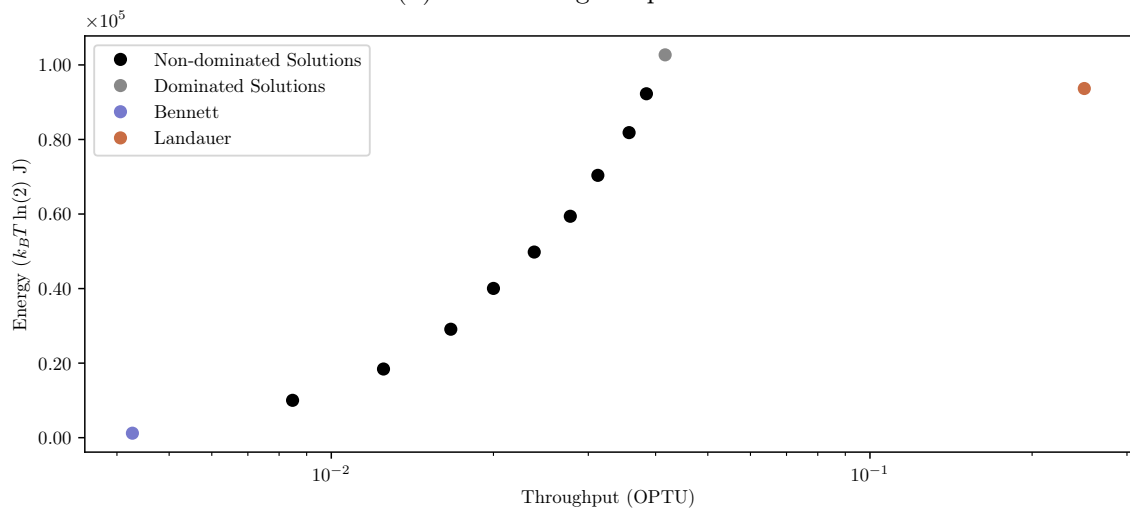
Figure 3.5: In-depth analysis of EPFL's Adder benchmark



(a) Cumulative energy loss

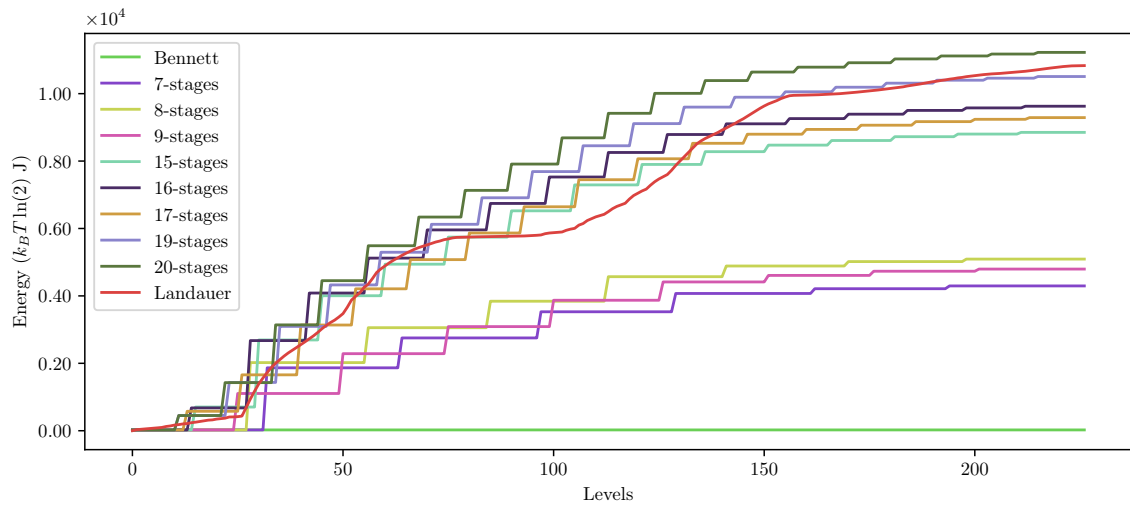


(b) Wires and gates per level

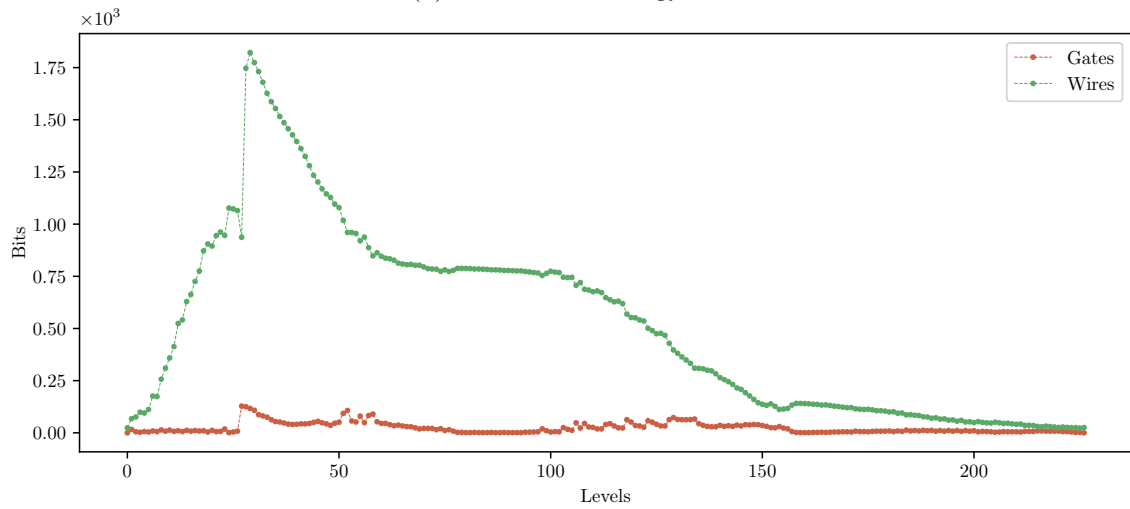


(c) Energy *versus* Throughput

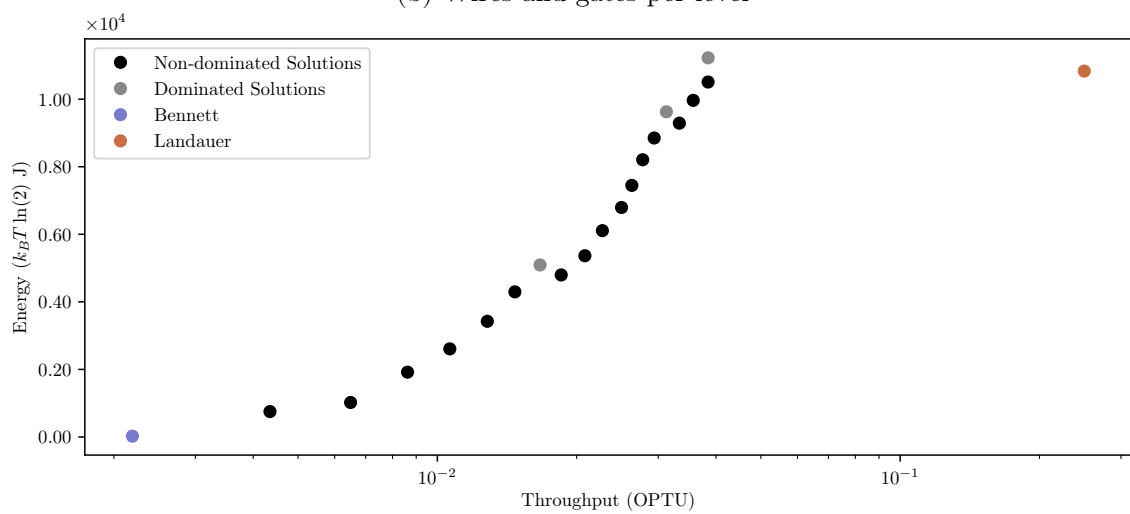
Figure 3.6: In-depth analysis of EPFL's Memory Controller benchmark



(a) Cumulative energy loss



(b) Wires and gates per level



(c) Energy *versus* Throughput

Figure 3.7: In-depth analysis of EPFL's Sin benchmark

patterns through the levels as seen in its Landauer’s curve. Near to the circuit’s input, it presents a linear loss rate (zone 0 to 25). In the following zones, it has a higher linear rate (zone 25 to 75), and then the rate decays (zone 75 to 100). Finally, it grows again (zone 100 to 150) and then decays (zone 150 to 225). This is a consequence of the variation in irreversible gates’ density along with the circuit levels as can be seen in Figure 3.7b. The Bennett’s configurations energy results, on the other hand, hide this information. The losses, in these cases, are similar to the pattern of the second group since the division in stages also accounts the cut wires that are no longer reversible. Note that for the sake of clarity, we do not present each possible, viable, partially reversible pipeline configuration in Figure 3.7a.

The wire’s density change in Sine’s circuit, showed in Figure 3.7b, create an exceptional situation: The expectation is that as the circuit is further divided, the throughput will improve and the energy will worsen. Nevertheless, this is not always the case as can be seen from the gray dots in Figure 3.7c that represent the dominated solutions. The 8-stages pipeline (the leftmost gray dot) has worse energy than the 9-stages pipeline. The same situation occurs when considering throughput, *i.e.*, the 9-stages pipeline is better than the 8-stages pipeline according to this other criteria. The 7-stages pipeline, compared with the 8-stages, presents better (smaller) energy. Hence, there is no reason to choose the 8-stages pipeline. In cases when energy is the priority, one must choose the 7-stages configuration. If the priority were throughput, the 9-stages pipeline must be chosen. The same situation occurs with 15, 16, and 17 stages. This similarity is due to both 8 and 16 stages, having a stage cut precisely in the wire’s density peak showed in Figure 3.7b. Dividing this circuit into 20 stages or more is not worth since its energy loss is worse than Landauer’s.

Table 3.1 summarizes the results. The first column presents the EPFL suite’s circuits. The second column displays the number of input/output bits and the third column shows the circuit’s depth. The fourth and fifth columns refer to Bennett’s clock compared to Landauer’s clock. Throughput Degradation presents the ratio between Landauer’s and Bennett’s throughput (less is better), and the Energy Improvement shows the ratio between Landauer’s and Bennett’s energy dissipation (more is better).

The last three columns are related to partially reversible pipelines. We show only the configuration with the maximum number of stages where the energy dissipation is smaller than the one in Landauer’s configuration (sixth column). Taking the Adder as an example, the 6-stages configuration has smaller energy than Landauer’s. The next configuration, *i.e.*, 7-stages, is energetically worse than Landauer’s (Figure 3.5a). Therefore, the 6-stages is the frontier for viable configurations of the Adder’s partially reversible pipeline. The Decoder entry in the table has these three columns empty

Table 3.1: Throughput and Energy evaluation results

Benchmark Name	I/O	Levels	Landauer over Bennett		Landauer over Partially reversible pipelines		
			Throughput Degradation	Energy Improvement	Number of stages	Throughput Degradation	Energy Improvement
Adder	256/129	255	128	8	6	22.0	1.09
Barrel shifter	135/128	12	<b>7</b>	<b>49</b>	6	2.0	1.06
Divisor	128/128	4470	2186	894	22	100.0	1.05
Hypotenuse	256/128	24801	12376	1674	71	175.5	1.01
Log2	32/32	444	<b>222</b>	<b>2004</b>	18	13.0	1.03
Max	512/130	287	144	11	11	14.0	1.06
Multiplier	128/128	274	<b>137</b>	<b>423</b>	32	5.0	1.01
Sine	24/25	225	<b>113</b>	<b>451</b>	19	6.5	1.03
Square-root	128/64	5058	2530	385	30	85.0	1.01
Square	64/128	250	<b>125</b>	<b>578</b>	74	2.5	1.01
Round-robin arbiter	256/129	87	<b>44</b>	<b>92</b>	5	9.5	1.02
Alu control unit	7/26	10	<b>6</b>	<b>50</b>	4	2.0	1.32
Coding-cavlc	10/11	16	<b>8</b>	<b>139</b>	8	2.0	1.03
Decoder	8/256	3	<b>2</b>	<b>76</b>	–	–	–
i2c controller	147/142	20	<b>10</b>	<b>18</b>	7	2.5	1.15
Int to float converter	11/7	16	<b>8</b>	<b>47</b>	9	1.5	1.01
Memory controller	1204/1231	114	<b>58</b>	<b>78</b>	10	6.5	1.02
Priority encoder	128/8	250	125	15	11	12.0	1.06
Lookahead XY router	60/30	54	28	9	8	4.0	1.04
Voter	1001/1	70	35	27	48	1.5	1.02
Average	225/143	1836	915	351	21	24.6	1.05



because it has only 3 levels. Thus, it is too narrow to be divided into stages.

Landauer clocked circuits always present better throughput values, and pure Bennett version (1-stage) always have fewer energy losses. Although, when we consider those two metrics simultaneously, interesting situations arise. In 60% of the circuits (12 in 20) the improvement in energy is greater than the degradation in the throughput (highlighted in bold).

If throughput metric is a concern in the design, then there are possibilities in the configuration space where the designer can reduce its degradation at the cost of increasing energy dissipation. For the EPFL benchmark, the designer has, in average, 21 different configurations for each of these circuits where he can trade between these metrics. The number of stages displayed for each circuit sets the boundary for the search in this configuration space. It is up to the designer to choose the best arrangement regarding its project requirements.

### 3.3 Algorithms

The previous section presented some interesting relations regarding energy and throughput in partially reversible pipelines. The energy  $e$  is directly related to the number of wires cut when dividing the circuit into stages. The throughput, on the other hand, is inversely proportional to the maximum stage size  $m$  (number of zones inside a stage). In this section, we present a strategy call lazy fanouts, which reduces the number of wires between stages. We also present two algorithms for splitting a Partially Reversible Pipelined QCA circuit into Bennett's stages. The algorithms focus on finding the set of indices where each Bennett's stage starts, while maintaining the energy dissipation and maximum stage size as low as possible. For the sake of brevity, we will denote the energy dissipation objective as  $Z(e)$  and the maximum stage size objective as  $Z(m)$ .

To be able to work with two objectives ( $Z(e)$  and  $Z(m)$ ), we used the concepts of *dominance*, *Pareto optimality*, and *Pareto frontier*. A given solution  $s_i$  *dominates* another solution  $s_j$  *if* one of  $s_i$ 's objectives is strictly better than its counterpart from  $s_j$  and all the others are better than or equal to their analogs in  $s_j$ . We also say that  $s_i$  is *Pareto-optimal if and only if* one of its objectives cannot be improved without degrading another one, i. e., there exists no solutions for the problem that dominates  $s_i$ . Finally, a *Pareto frontier* from a given set  $S$  of solutions is the subset  $F \subseteq S$  of solutions non-dominated by any other in  $S$ .

The first algorithm, called the Section-Splitter Algorithm, SSA, is used as base-

line. It is an heuristic that finds good  $(Z(e), Z(m))$  points when receiving a fixed number of stages as input. The second, called the Optimal splitter algorithm, OSA, is the main contribution of this work. Instead of dealing with a fixed number of stages, it receives as input a maximum stage size restriction, i. e., the maximum value allowed for  $Z(m)$ . Then, for the given restriction, it finds the division indices that yield the Pareto-optimal point  $(Z(e), Z(m))$ , where  $Z(e)$  is the minimum possible. We also show an extension that does the opposite, i. e., given a maximum energy restriction, it finds the division indices that yield the Pareto-optimal point  $(Z(e), Z(m))$ , where  $Z(m)$  is the minimum possible.

### 3.3.1 Lazy Fanouts Netlist Manipulation Strategy

To understand how our *lazy fanouts* strategy works, one should first understand how does the netlist manipulation used in Section 3.2. We call their strategy as *early fanouts* since it expands every fanout at the origin gate's level. It effectively creates all needed wires at once and routes them to their destinations. Figure 3.8a shows how does a netlist looks when using *early fanouts*.

Despite making the representation simpler and resembling the graph, the *early fanouts* strategy has some drawbacks when employed in Partially Reversible Pipelines, since it increases the number of wires passing through every level. In this work, we apply a different netlist manipulation strategy. We call our strategy as *lazy fanouts*, and it only expands a fanout right before the target gate. Hence, it avoids passing wires through levels before they are needed, thus reducing the energy dissipation. Figure 3.8b shows how does the same netlist in Figure 3.8a looks when manipulated using *lazy fanouts*.

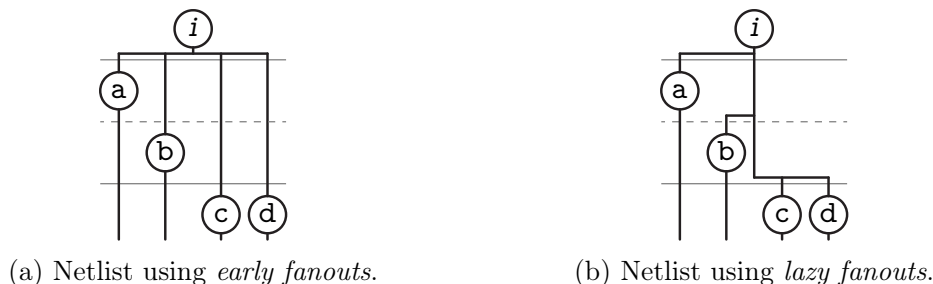


Figure 3.8: Netlist manipulation methods.

Comparing both strategies in practice (Figures 3.8a and 3.8b), one can see that when we divide the circuit on the dashed level, the *early fanouts* causes the dissipation

of 4 bits of energy while the *lazy fanouts* strategy dissipates 3 bits, reducing the energy dissipation by 25% with no effect on throughput.

We carried out an experiment to evaluate the energy reduction by using the *lazy fanouts* strategy over the *early fanouts*. To be fair when comparing both strategies, we considered that every circuit uses every level as a section. This way, we can compare how each strategy affects the energy in the worst-case scenario, i. e., the one with maximum energy dissipation.

Table 3.2 presents the results of this experiment, where the first column displays the benchmark name, the second the circuit's depth, and the third the average energy reduction. We evaluated the reduction for each circuit as  $1 - l_e/e_e$ , where  $l_e$  and  $e_e$  are the energy from the lazy and early strategies, respectively. One can see that manipulating the netlist reduced the energy dissipation by 47.30% on average. As in the previous analysis, we found better results for deeper over shallower circuits. This disparity happens because in deeper circuits the *early fanouts* strategy needs to spawn a more significant number of wires since gates usually are more distant.

### 3.3.2 The Section-Splitter Algorithm

Our algorithm receives as input a list  $B$  of bit counts passing through each one of the  $n$  levels, a target number of divisions  $s$ , and a maximum difference between two section sizes  $d$ . It works by creating ranges where it can choose a position to start a new section while respecting the  $d$ 's constraint. Then, for each range  $R$ , it selects the position  $p \in R$  that will dissipate the minimum energy, i. e., where  $B_p$  is minimal.

We present the section-splitter's pseudocode in Algorithm 2. Line 1 creates the set *cuts* that will hold the start positions of each section. It always selects the index 0, since it marks the start of the circuit. Line 2 initializes the first exact cut (*pos*), and line 3 iterates it over all possible exact cuts on  $B$ . Lines 4 and 5 select the range whence the algorithm will select the cut. Each range starts at  $-d/4$  from the exact cut and ends at  $d/4$  after it, ensuring that the difference between two sections will never be bigger than  $d$ , as proved below. Lines 6 and 7 selects the position with minimum bits (energy) in the range and save it to the *cuts* set. Line 8 changes *pos* to the next exact cut. Finally, line 9 returns the set of chosen cuts.

To demonstrate that the section-splitter algorithm never selects two sections with size difference bigger than  $d$ , we will consider the worst case scenario. In this scenario, a section  $S_1$  begins at the stop of a range centered at  $pos_1$  and ends at the start of the next one (centered at  $pos_1 + n/s$ ), i. e., its size is the minimum possible. Besides, a different section  $S_2$  begins at the start of a range centered at  $pos_2$  and ends at the

Table 3.2: Energy reduction from using *lazy fanouts*.

Benchmark	Levels	Reduction %
<i>adder</i>	255	19.20
<i>arbiter</i>	87	88.14
<i>bar</i>	12	51.77
<i>cavlc</i>	16	34.65
<i>ctrl</i>	10	42.09
<i>dec</i>	3	0.00
<i>div</i>	4372	91.50
<i>hyp</i>	24 801	93.54
<i>i2c</i>	20	24.03
<i>int2float</i>	16	28.75
<i>log2</i>	444	86.85
<i>max</i>	287	1.80
<i>mem_ctrl</i>	114	65.67
<i>multiplier</i>	274	58.15
<i>priority</i>	250	46.52
<i>router</i>	54	23.04
<i>sin</i>	225	74.87
<i>sqrt</i>	5058	86.56
<i>square</i>	250	18.42
<i>voter</i>	70	10.47
<b>Average</b>	1831	47.30

stop of the next one (centered at  $pos_2 + n/s$ ), i. e., its size is the maximum possible. Therefore,  $S_1$  begins at  $start_1 \leq (pos_1 + d/4)$  and ends at  $end_1 \geq (pos_1 + n/s - d/4)$ , thus its minimum size is  $min_1 = n/s - d/2$ . Furthermore,  $S_2$  begins at  $start_2 \geq (pos_2 - d/4)$  and ends at  $end_2 \leq (pos_2 + n/s + d/4)$ , thus its maximum size is  $max_2 = n/s + d/2$ . One can see that the maximum size difference between  $S_1$  and  $S_2$  is  $|min_1 - max_2| = d$ .

We created an example to illustrate how the section-splitter algorithm works. Figure 3.9a shows the input list  $B$ , sized  $n = 10$ , storing the number of bits passing through a circuit. Also, we set  $s = 4$  and  $d = 2$ . Figure 3.9b shows the solution returned by the naïve approach, i. e., to divide on the *floor* of each exact division, where each dashed line denotes a division. One can see that its throughput is  $1/8$  since its max section is  $m = 3$ . Also, it dissipates  $8+3+7+3 = 21$  bits of energy. Now, let us consider the section-splitter algorithm. Figure 3.9c shows each range of possible division created by it. For each range, it selects the position that dissipates the minimum energy, resulting in the division on Figure 3.9d. One can see that its throughput is also  $1/8$ ,

**Algorithm 2:** The Section-Splitter Algorithm.

---

**input** :  $B \leftarrow$  list of size  $n$  of bits passing through each level, including circuit's inputs and outputs  
**input** :  $s \leftarrow$  target amount of stages  
**input** :  $d \leftarrow$  maximum stage size difference allowed  
**output**:  $divs \leftarrow$  set of indices limiting the stages  
1  $divs \leftarrow \{0\}$   
2  $pos \leftarrow n/s$   
3 **while**  $pos < n$  **do**  
4      $sta \leftarrow \lceil pos - d/4 \rceil$   
5      $sto \leftarrow \lfloor pos + d/4 \rfloor$   
6      $div \leftarrow \arg \min_{x \in [sta..sto]} (B_x)$   
7      $divs \leftarrow divs \cup \{div\}$   
8      $pos \leftarrow pos + n/s$   
9 **return**  $divs$

---

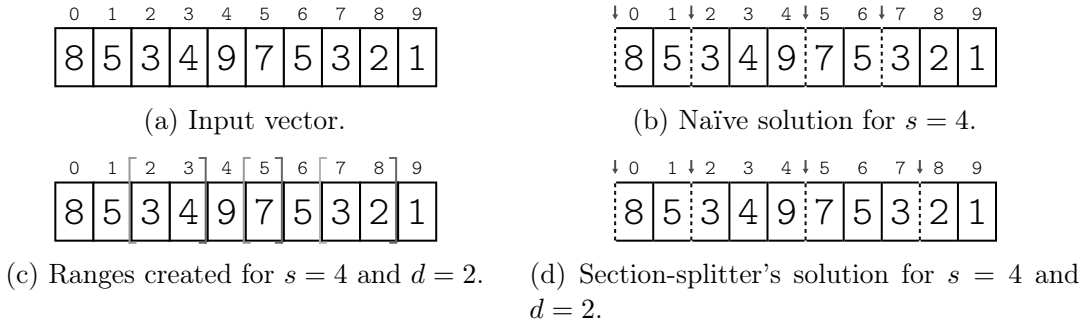


Figure 3.9: Section-splitter example.

but it dissipates  $8 + 3 + 7 + 2 = 20$  bits of energy. Therefore, the algorithm was able to reduce the energy dissipation without affecting the throughput. Note that this is not always the case since the algorithm prioritizes energy over throughput.

In this experiment, we compare the naïve division algorithm results showed in Section 3.2 ( $alg_n$ ) with the proposed section-splitter algorithm ( $alg_s$ ). To show all the extra possibilities that  $alg_s$  can find, we explored its entire search-space, i. e., we ran it for every possible section number  $s$ , and maximum difference between section sizes  $d$ . Therefore, for a benchmark with depth  $n$ , we tested  $s \in [2, n]$ , since  $s = 1$  means Bennett's clocking and  $s > n$  is impossible, and  $d \in [2, \lfloor 2^n/s \rfloor]$ , since  $d = 1$  means the naïve division and  $d > 2^n/s$  may lead to range overlapping.

Aside from the figures, we numerically evaluated each pair  $(b, m)$  of benchmark and netlist manipulation strategy. To achieve this, we joined the solutions from the Pareto frontier of  $alg_n$  ( $F_1$ ) and  $alg_s$  ( $F_2$ ) for  $(b, m)$  and evaluated the Pareto frontier

of  $F_1 \cup F_2$  ( $F_{1,2}$ ). Then, we counted the percentage  $p_1$  and  $p_2$  of solutions from  $F_1$  and  $F_2$ , respectively, on  $F_{1,2}$ . This way, one can infer the percentage of solutions of  $alg_n$  that were dominated by  $alg_s$  by calculating  $100\% - p_1$  and vice-versa. For example, consider two algorithms  $alg_i$  and  $alg_j$ . If the value of  $p_i$  for a given pair  $(b, m)$  is 30%, it means that 70% of  $F_i$ 's solutions were dominated by  $F_j$ 's solutions.

Table 3.3: Section-splitter algorithm *vs.* the naïve approach.

Benchmark	Levels	<i>Early fanouts</i>		<i>Lazy fanouts</i>	
		alg <sub>n</sub> %	alg <sub>s</sub> %	alg <sub>n</sub> %	alg <sub>s</sub> %
<i>adder</i>	255	16.67	100.00	14.29	100.00
<i>arbiter</i>	87	0.00	100.00	60.00	100.00
<i>bar</i>	12	60.00	87.50	60.00	71.43
<i>cavlc</i>	16	66.67	100.00	66.67	100.00
<i>ctrl</i>	10	100.00	100.00	100.00	100.00
<i>dec</i>	3	100.00	50.00	100.00	50.00
<i>div</i>	4372	31.82	99.91	20.20	96.90
<i>hyp</i>	24801	11.27	99.85	32.44	93.69
<i>i2c</i>	20	16.67	100.00	16.67	100.00
<i>int2float</i>	16	66.67	100.00	66.67	100.00
<i>log2</i>	444	5.56	100.00	38.89	89.26
<i>max</i>	287	0.00	100.00	0.00	100.00
<i>mem_ctrl</i>	114	20.00	100.00	18.75	98.59
<i>multiplier</i>	274	25.00	100.00	27.59	99.32
<i>priority</i>	250	63.64	100.00	57.89	100.00
<i>router</i>	54	71.43	100.00	62.50	97.22
<i>sin</i>	225	25.00	98.58	29.17	92.05
<i>sqrt</i>	5058	40.00	99.69	30.19	95.46
<i>square</i>	250	51.85	100.00	53.57	100.00
<i>voter</i>	70	76.92	92.11	61.54	97.14
<b>Average</b>	1831	42.46	96.38	45.85	94.05

Table 3.3 shows the results of this experiment. The first column presents the benchmark name, and the second column its depth. The next two columns show, respectively, for the netlists employing *early fanouts*, the percentage of solutions on  $F_{1,2}$  from  $F_1$  and  $F_2$ . The two final columns show this same data, but for the netlists employing *lazy fanouts*.

This data shows that our algorithm's solutions dominate about 56% of solutions from the naïve approach while being about only 5% dominated. Besides, it works better on deeper circuits than on shallower ones. On deep circuits, e. g., *div* and *hyp*,

our algorithm can explore better the configuration space since there is plenty of space to find a good place to start a new section. On the other hand, shallow circuits, such as the *dec* and the *ctrl*, do not permit a more significant variation on the  $d$  parameter. Therefore, our algorithm cannot substantially improve the circuit's energy dissipation in these cases.

There is also a slight difference in our algorithm's performance on circuits employing *early fanouts* and on circuits using *lazy fanouts*. Since the lazy strategy passes a smaller number of wires through different levels, it gives to our algorithm a smaller margin to modify the division points. Thus, circuits using the *early fanouts* method have more flexibility for optimizations.

We created an example to illustrate how the SSA works. Fig. 3.9a shows the input list  $B$ , sized  $n = 10$ . Also, we set  $s = 4$  and  $d = 2$ . Fig. 3.9c shows each range of possible division indices created by the SSA and Fig. 3.9d shows its resulting division. One can see that it has maximum stage  $Z(m) = 3$  and dissipates  $Z(e) = 8 + 3 + 7 + 2 = 20$  bits of energy.

### 3.3.3 The Optimal Splitter Algorithm

The Optimal Splitter Algorithm (OSA) receives as input a list  $B = \{b_0, \dots, b_{n-1}\}$  of bit counts passing through each one of the  $n$  levels, and a bound  $m$ , representing the maximum  $Z(m)$  allowed. Then, it finds the division indices that yield the Pareto-optimal point  $(Z(e), Z(m))$  where  $Z(e)$  is minimum and  $Z(m) \leq m$ . This is done by representing the bits passing through the circuit's levels as a graph and solving a modified bi-objective shortest path problem on it.

#### 3.3.3.1 Graph Representation

Given an input vector  $B'$ , where  $B' = B \cup \{0\}$ , the OSA constructs a Directed Acyclic Graph (DAG)  $G = (V, A)$  as follows. Let  $V$  be a set of vertices and  $A$  be a set of arcs. We define a bijective function  $f : V \mapsto B'$  that associates each vertex  $v_i \in V$  with the input  $b_i \in B'$ . Furthermore, each vertex  $v_i \in V$  associates with an energy value  $e_i = b_i$ . Given the maximum stage size  $m$ , we construct  $A$  as follows. For each vertex  $v_i \in V$ , we define an arc  $(v_i, v_{i+j})$  for all  $1 \leq j \leq m$  and  $i + j \leq |V|$ . In addition, each arc  $(v_i, v_j) \in A$  has an associated weight  $w_{i,j} = j - i$ . Fig. 3.10a shows the resulting DAG for the input vector given on Fig. 3.9a when  $m = 2$ .

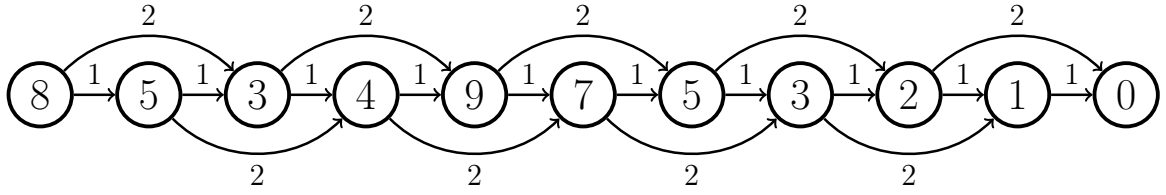
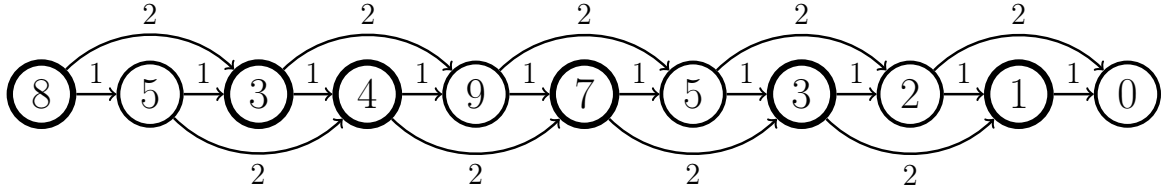
(a) DAG constructed by OSA for  $m = 2$ .(b) OSA's solution for  $m = 2$ . The selected vertices, i. e., divisions, are shown in bold.

Figure 3.10: Optimal Splitter Algorithm example.

### 3.3.3.2 Minimum Energy Given Maximum Stage Size

Note that returning  $Z(e)$  alone and using  $Z(m) = m$  does not guarantee that the solution is Pareto-optimal, as a smaller  $m$  may yield the same  $Z(e)$ . Therefore, we developed a bi-objective adaptation from the shortest path algorithm for DAGs that uses the energy values  $e_i$  as cost. As a second objective, it also minimizes  $Z(m)$  for the  $Z(e)$  found. This is done by preferring (i) the paths with a smaller  $Z(e)$ , i. e., the default for the shortest path algorithm; or (ii) a smaller  $Z(m)$  when there are two choices with equal  $Z(e)$ .

One can see that the maximum  $Z(m)$  restriction is automatically respected, since there are no arcs connecting two vertices  $v_i, v_j$ , where  $j - i > m$ . Hence  $v_i, v_j$  will never be subsequent on the selected path. Also, adding preference (ii) will not change the correctness of the OSA as a shortest path solver. Since its clause requires that  $Z(e)$  is equal to the minimum found, the algorithm will keep the behavior of choosing paths with lower costs. Fig. 3.10b presents the solution given by OSA for the DAG shown in Fig. 3.10a. It has maximum stage  $Z(m) = 2$  and dissipates  $Z(e) = 8 + 3 + 4 + 7 + 3 + 1 = 26$  bits of energy.

Algorithm 3 shows how OSA works. It starts by initializing the vertices' best objectives found as infinite (Lines 1 to 3). Then, it initializes those objectives for the source vertex, i. e., the first division index (Lines 4 and 5). Lines 6 to 15 are the shortest path's main evaluation loop. Since we are dealing with a DAG, we can use a simplified shortest path algorithm that iterates in topological order. Line 6 iterates on



**Algorithm 3:** The Optimal Splitter Algorithm

---

```

input :  $G \leftarrow$  graph representation
output:  $divs \leftarrow$  set of indices limiting the stages
1 for  $i \leftarrow 0$  to  $|V| - 1$  do
2    $Z(e)_i \leftarrow \infty$ 
3    $Z(m)_i \leftarrow \infty$ 
4  $Z(e)_0 \leftarrow e(0)$ 
5  $Z(m)_0 \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|V| - 1$  do
7   for  $a_{i,j} \in \text{arcs}(v_i)$  do
8      $new_e \leftarrow Z(e)_i + e(j)$ 
9      $new_m \leftarrow \max\{w_{i,j}, Z(m)_i\}$ 
10     $bet_e \leftarrow new_e < Z(e)_j$ 
11     $bet_m \leftarrow (new_e = Z(e)_j) \wedge (new_m < Z(m)_j)$ 
12    if  $bet_e \vee bet_m$  then
13       $Z(e)_j \leftarrow new_e$ 
14       $Z(m)_j \leftarrow new_m$ 
15       $parent(j) \leftarrow i$ 
16  $divs \leftarrow \emptyset$ 
17  $c \leftarrow |V| - 1$ 
18 while  $c \neq 0$  do
19    $c \leftarrow parent(c)$ 
20    $divs \leftarrow \{c\} \cup divs$ 
21 return  $divs$ 

```

---

all vertices  $v_i \in G$ , starting at  $v_0$ , and Line 7 iterate on  $v_i$ 's arcs. Lines 8 and 9 evaluate the new  $Z(e)$  and  $Z(m)$  for  $v_j$  if the path to  $v_j$  passes through  $v_i$ . Lines 10 to 12 check if preferences (i) or (ii) are fulfilled. If true, Lines 13 to 15 update both objectives for  $v_j$  and its parent. Lines 16 to 20 builds the division indices by following the shortest path from  $v_0$  to  $v_n$  in reverse order. Note that  $v_n$  is discarded. Finally, Line 21 returns the division indices. OSA's complexity is  $\mathcal{O}(n^2)$ . We prove the optimality of OSA on Theorems 1 and 2.

**Lemma 1** *The OSA always returns the division indices that yield the minimum  $Z(e)$ .*

**Proof.** We know that, as a shortest path solver, the OSA gives us the set of vertices  $\mathcal{V} \subseteq V$  that form the shortest path from  $v_0$  to  $v_n$ , i. e.,  $\sum_{v_i \in \mathcal{V}} e_i$  is minimum. Therefore, the indices represented by the selected vertices are the ones who will yield the minimum  $Z(e)$ .  $\square$

**Lemma 2** *The OSA always returns the cuts that yield the minimum  $Z(m)$  for the  $Z(e)$  found.*

**Proof.** Basing on Theorem 1 and knowing, by preference (i), that the final  $Z(m)_i$  for a given vertex  $v_i$  will only be defined when then minimum  $Z(e)_i$  is found, we consider, without loss of generality, that the OSA will only process the  $Z(m)$  on the paths with minimum  $Z(e)$ . Therefore, for the  $Z(m)$ , the problem becomes an instance of the minimax path problem, i. e., to find a path from  $v_0$  to  $v_n$  where the maximum weighted arc selected is the minimum possible.

One can solve the minimax path problem through an adaptation of the shortest path problem that chooses paths with lower maximum arcs' weights. Since we are assuming that the  $Z(e)$  is equal for all paths, such adaptation is achieved by preference (ii). Therefore, the OSA solves the minimax path problem on these minimum  $Z(e)$  paths, hence being able to find the Pareto-optimal point  $(Z(e), Z(m))$ .  $\square$

### 3.3.3.3 Smallest Maximum Stage Size Given Energy

As proved on Theorem 2, the OSA always finds the divisions that yield the Pareto-optimal point  $(Z(e), Z(m))$ , where  $Z(e)$  is minimum, given a maximum  $Z(m)$  restriction. One might want to go the other way ahead, i. e., finding the divisions indices' that yield the minimum  $Z(m)$  and  $(Z(e), Z(m))$  is Pareto-optimal, given a maximum  $Z(e)$  restriction. Hence, based on Theorem 3, we developed an iterative version from the OSA that solves this problem.

**Lemma 3** *Consider two solutions returned by the OSA,  $a = (Z(e)_a, Z(m)_a)$  and  $b = (Z(e)_b, Z(m)_b)$ , found using the same list of bits  $B$ , but with maximum  $Z(m)$  equal to  $m_a$  and  $m_b$ , respectively. Then,  $Z(e)_a \leq Z(e)_b \iff m_a \geq m_b$ .*

**Proof.** First, we define  $G_a$  and  $G_b$  as the graphs constructed by the OSA for  $a$  and  $b$ , respectively. To prove the *if* part, assume that  $m_a < m_b$ , which implies that  $G_a \subset G_b$ . This means that all paths in  $G_a$  are in  $G_b$  and yet  $G_b$ 's shortest path is larger than  $G_a$ 's. Therefore, we have a contradiction. By the same reason, it is not possible that  $Z(e)_a > Z(e)_b$  when  $G_b \subseteq G_a$ . Hence, by contradiction, the *only if* part is also true.  $\square$

Given the fact that the points generated by the OSA are always Pareto-optimal, we know that, if we apply it for all possible maximum stage sizes restrictions  $m$ , i. e.,  $m \in [1..n]$ , we will generate all Pareto-optimal points. Hence, basing on Theorem 3, we know that the  $Z(e)$  of these points will be on descending order when  $m$  is crescent.

Therefore, given a maximum energy restriction, we can make a binary search on these OSA's results and generate only the ones required through the search. Hence, we can solve the problem in  $\mathcal{O}(n^2 \log_2(n))$  time.

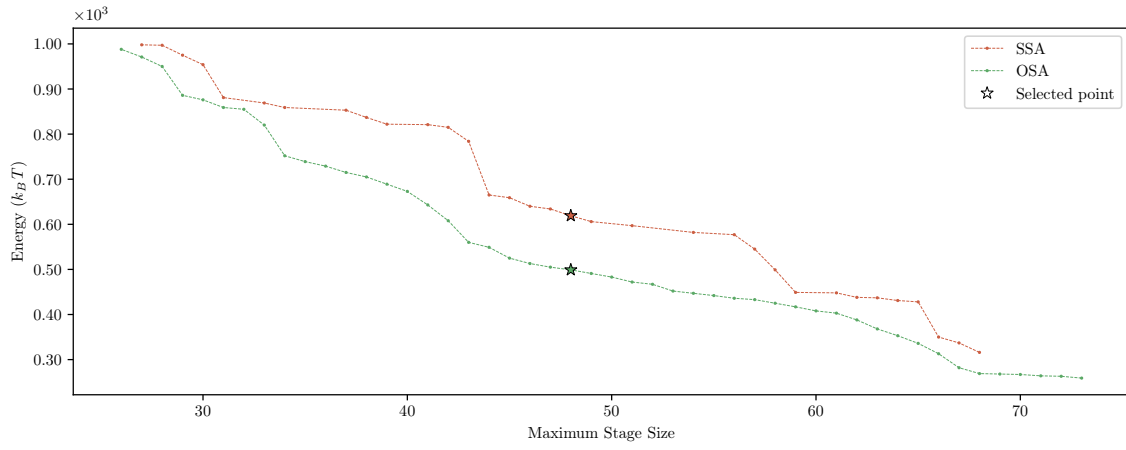
This section compares the effectiveness of the OSA over the SSA. We ran the OSA for all possible maximum stage sizes  $m$ , i. e.,  $m \in [1..n]$ , where  $n$  is the circuit's depth. To make a fair comparison, we also ran the SSA for its entire search-space, i. e., for every possible stages number  $s$ , and maximum difference between stage sizes  $d$ . Therefore, we evaluated  $s \in [1..n]$ , and  $d \in [2.. \lfloor 2n/s \rfloor - 1]$ . Note that  $d \leq 1$  causes empty ranges, and  $d \geq 2n/s$  leads to range overlapping. All netlists employed the *lazy fanouts* strategy, i. e., the fanouts were delayed to reduce the number of wires passing through stages.

Fig. 3.11 shows an in-depth analysis of the *sin* circuit. First, Fig. 3.11a shows part of the Pareto-frontier around some selected points. One can see that, in this area, the OSA's points dominate all of SSA's points. To select those points, we defined a maximum  $Z(m)$  restriction  $m = 48$  and chosen the point with best energy from the SSA ( $Z(e) = 619, Z(m) = 48$ ) and the OSA ( $Z(e) = 499, Z(m) = 48$ ), under this restriction (marked by stars). Fig. 3.11b shows the selected division indices from both points. Some of SSA's indices are stuck in local minima, which caused that, even with the SSA selecting less indices than the OSA (5 and 6, respectively), it dissipated more energy. Fig. 3.11c shows the cumulative energy sum per level. Despite that SSA achieves a lower energy on some levels, the OSA achieves the minimum final energy dissipation.

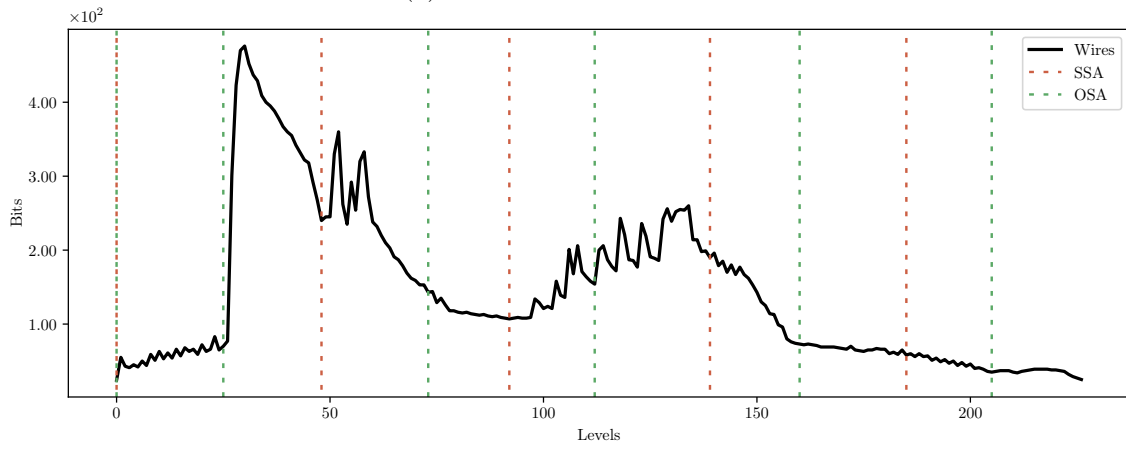
Aside from the figures, we numerically evaluated each benchmark. To achieve this, we joined the solutions from the Pareto frontier of SSA ( $F_s$ ) and OSA ( $F_o$ ) for a given benchmark, i. e.,  $F_s \cup F_o$ , and then we evaluated the general Pareto frontier ( $F_g$ ) from the union. Then, we counted the percentage  $p_s$  and  $p_o$  of solutions from  $F_s$  and  $F_o$ , respectively, on  $F_g$ . Thus, one can infer the percentage of solutions of SSA that were dominated by OSA by calculating  $100\% - p_s$  and vice-versa. For example, consider two algorithms  $alg_i$  and  $alg_j$ . If the value of  $p_i$  is 30%, it means that 70% of  $F_i$ 's solutions were dominated by  $F_j$ 's solutions.

Table 3.4 shows the results of this experiment. The first column presents the benchmark name, while the second column shows the number of levels, including the inputs and outputs, of each benchmark. The third and fourth column present, respectively, the number of points of the OSA in  $F_g$  and the percentage of the points generated by OSA that are non-dominated within  $F_g$ . The two remaining columns show the same information for the SSA.

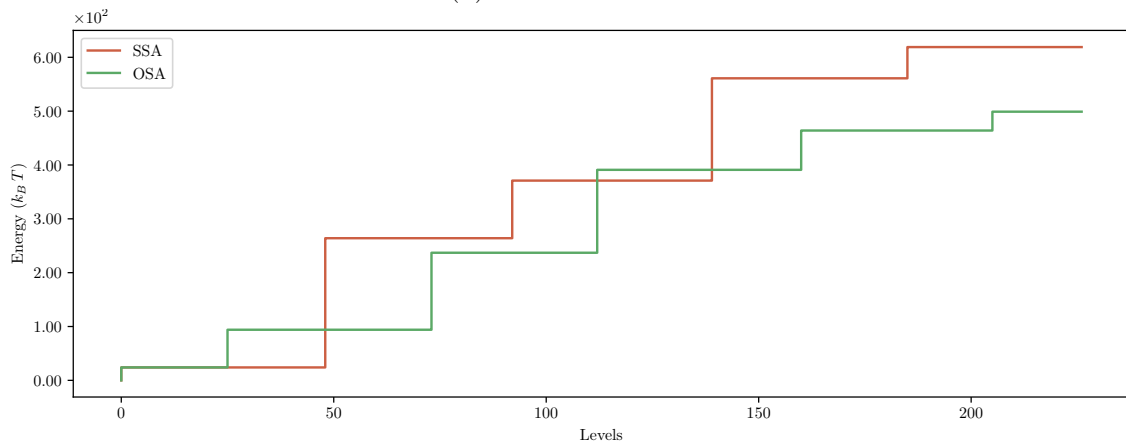
One can see from Table 3.4 that all OSA's solutions are in  $F_g$ . Furthermore,



(a) Zoomed Pareto-frontier.



(b) Division indices.



(c) Cumulative energy.

Figure 3.11: In-depth analysis of *sin* benchmark: SSA vs. OSA.

Table 3.4: OSA *vs.* SSA.

Benchmark	Levels	OSA		SSA	
		Points	Percent	Points	Percent
<i>adder</i>	257	194	100.00	37	31.90
<i>arbiter</i>	89	28	100.00	12	70.59
<i>bar</i>	14	11	100.00	5	71.43
<i>cavlc</i>	18	18	100.00	8	72.73
<i>ctrl</i>	12	12	100.00	6	85.71
<i>dec</i>	5	5	100.00	2	66.67
<i>div</i>	4374	953	100.00	40	7.27
<i>hyp</i>	24803	2926	100.00	84	6.41
<i>i2c</i>	22	22	100.00	8	57.14
<i>int2float</i>	18	18	100.00	8	72.73
<i>log2</i>	446	250	100.00	8	6.50
<i>max</i>	289	261	100.00	50	27.32
<i>mem_ctrl</i>	116	103	100.00	28	40.00
<i>multiplier</i>	276	220	100.00	61	41.50
<i>priority</i>	252	163	100.00	47	44.34
<i>router</i>	56	54	100.00	20	57.14
<i>sin</i>	227	144	100.00	19	21.11
<i>sqrt</i>	5060	937	100.00	67	13.81
<i>square</i>	252	190	100.00	49	42.61
<i>voter</i>	72	59	100.00	15	40.54
<b>Average</b>	1833	328	100.00	29	43.87

OSA generates an average of 328 points on the Pareto frontier, such that circuits with a higher number of levels also have a larger number of points on their Pareto frontiers. On the other hand, SSA only have an average of 29 points into the Pareto frontier, which represents a total of 45.87% of the points generated by this algorithm, on average. We highlight that the points generated by the SSA that are in the Pareto frontier were also generated by the OSA. Therefore, those results indicate that OSA greatly outperforms SSA. The proposed algorithm generates the whole Pareto frontier, encompassing all the points given by the SSA. Furthermore, it generates a larger set of points, which gives a wider set of options to the circuit designer to select a preferred energy dissipation / throughput ratio.

## 3.4 Summary

In this chapter, we proposed a technique that explores the optimal energy vs. throughput tradeoff in Partially Reversible Pipelined QCA circuits. OSA divides a QCA circuit into Bennett's stages with the minimum possible energy dissipation while considering a minimum throughput restriction. We also propose an extension that does the opposite, i. e., it gives the maximum throughput considering the maximum energy restriction. We also prove that both techniques are exact. These algorithms provide circuit designers with a better and larger set of possible configurations to use.

# Chapter 4

## Partial Reversibility via Layout

Chapter 3 showed strategies to reduce energy in FCN circuits by clocking modifications. The advantage is that no layout change was needed, despite the degradation of the circuit throughput. Here, we take a different approach based on layout changes instead of modifications on timing. We propose an intermediary step towards fully reversible systems by introducing a post-synthesis strategy that reduces fundamental energy losses. Our main contribution is a novel technique that identifies exploitable fan-outs and uses them for the recovery of bit energy in FCN circuits, thus enabling the design of new types of *partially reversible systems*. This is done by embedding fan-outs into a gate, creating what we call *bit recycling gates*. Moreover, we propose an algorithm that creates partially reversible systems while allowing the designer to choose between energy reduction with no effect on delay (*depth-oriented*), or to focus solely on energy and accept a potential delay penalty (named *energy-oriented*).

### 4.1 Energy Evaluation in FCN Circuits

The main motivation for designing reversible systems is to ensure energy scalability, i. e., to guarantee that also in next technology generations energy reduction is possible. Therefore, it is important to quantify any energy losses of the system, ideally including both fundamental and non-fundamental ones. In case of the QCA technology, QCAPro is the first tool that enabled the energetic evaluation of QCA designs [Srivastava et al., 2011]. It applies a model that estimates polarization errors within the QCA cells and energy loss in non-adiabatic switching operations. Hence, the QCAPro provides an upper-bound energy consumption.

Torres et al. [2018] recently proposed a logical synthesis model for QCAs that incorporates the main physical aspects, such as energy dissipation. The authors also

developed a tool, called QCADesigner-E, that performs the energy evaluation in QCA circuits. The tool uses a quantum-level model for QCA cells and technological parameters such as cell size, cell geometry, the electrical permittivity of material for QCA system. It enables the energy evaluation that arises in irreversible as well as in reversible elements like interconnection, such as wires and fan-outs. Consequently, the QCADesigner-E provides the total energy consumed by a QCA circuit with specific technological parameters.

Ercan and Anderson provide a method that solely extracts the fundamental losses and disregards any non-fundamental ones [Ercan and Anderson, 2013; Anderson, 2013]. Their method is grounded in physical information theory, which guarantees generality to their result, i.e., it is genuinely fundamental energy lower bound. As the authors said, this bound reflects the local physical consequences of the logic gate's cells interacting with a heat bath given the global constraints imposed by quantum dynamics and thermodynamics. Therefore, this is an appropriate approach for studies focusing on the energy scalability of the designs, having in mind that non-fundamental losses can be reduced by technological advances, while fundamental losses must be treated in a different way [Li and Vitanyi, 1996].

As our main focus is Landauer clocking, we use Ercan and Anderson's energy evaluation framework, which is different than our Bennett's clocking energy evaluation. We discuss about those differences and the issues related to an unified energy evaluation approach in Chapter 5.

In order to compute a gate's unavoidable fundamental energy dissipation, one should calculate the difference between the entropies of the probabilities of the initial and final states of the gates. The Equations 4.1a and 4.1b presents the Shannon's entropy for a generic probability distribution  $\mathcal{S}$ . In many cases there is no knowledge about the probability  $P(x_i)$  of each of its initial states  $x_i \in X$ , thus, although the formalism does not requires it, each combination is considered to be equiprobable. Given  $P(x_i)$ , we may now define the probability  $P(y_i)$  of each of the gate's final states  $y_i \in Y$ . It is achieved by applying the gate's logic function  $\mathcal{G}$  on each  $x_i$  and adding  $P(x_i)$  to  $P(\mathcal{G}(x_i))$ . Then, by calling Eq. (4.1a) on  $x_i$ , we evaluate the contribution  $\mathcal{I}(x_i)$  that it adds to the overall Shannon's entropy of  $X$ . Hence, the entropy  $\mathcal{H}(X)$  of  $X$  is the summation of the contributions of all initial states  $x_i$  (Eq. (4.1b)). The same is valid for  $Y$ , i.e.,  $\mathcal{H}(Y)$  is the summation of the contributions  $\mathcal{I}(y_i)$  of all final states  $y_i$ . The Landauer's limit for this gate is the difference between Shannon's entropy of its initial states and final states, i.e.,  $\mathcal{H}(X) - \mathcal{H}(Y)$ . Finally, the Landauer's limit of the circuit is the sum of all losses of the gate. Following example shall detail this energy



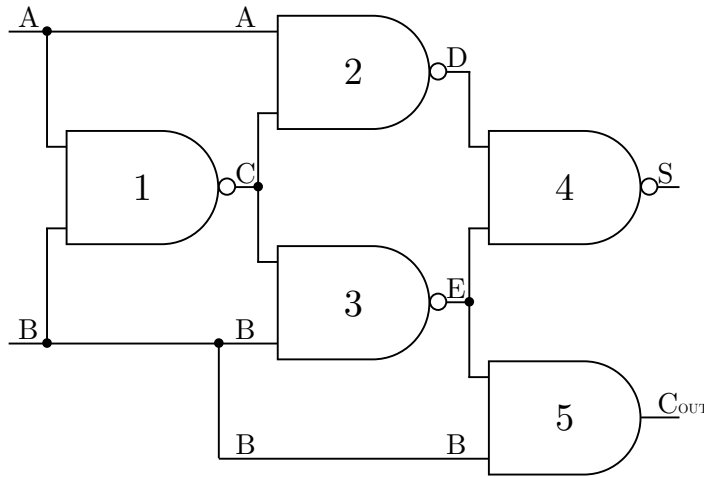


Figure 4.1: Half-adder circuit.

assessment.

$$\mathcal{I}(s) = -P(s) \times \log_2(P(s)) \quad (4.1a)$$

$$\mathcal{H}(S) = \sum_{s \in S} \mathcal{I}(s) \quad (4.1b)$$

**Example 1** Consider the half-adder circuit depicted in Fig. 4.1. As the NAND gate 1 receives two of the four primary inputs of the circuit, each of its possible input combinations has a probability of  $1/4$ . Therefore, the Shannon's entropy of its initial state is  $\mathcal{H}(X) = 2$  bits, as listed in Table 4.1a. Replicating this calculus for the final states of gate 1 (see also Table 4.1a), the Shannon's entropy of the final state is  $\mathcal{H}(Y) = 0.81$  bits. That means, gate 1 loses  $\mathcal{H}(X) - \mathcal{H}(Y) = 1.19$  bits of information.

Table 4.1b presents the same calculation for gate 2. Despite this gate being a NAND like gate 1, it has distinct fundamental losses due to their placement in the circuit, i.e., both gates operate with different initial states' probability (one of gate 2 input bits is produced by gate 1 and other is a primary input). In the case of gates 3 and 5, as the initial and final states' probabilities are the same as those in gate 2, both dissipate the same amount of energy (Tables 4.1c and 4.1e). Although, this is not the case for gate 4, which loses less information (Table 4.1d).

Finally, adding up all gate's losses gives us the Landauer's limit for this half-adder circuit. Hence, we conclude that this circuit loses 3.76 bits of information or has a fundamental energy dissipation limit of  $3.76 k_B T \ln(2)$  Joules.

Table 4.1: Entropy calculations for gates in the half-adder of Fig. 4.1

(a) Gate 1.								(b) Gate 2.											
Input				Output				Input				Output							
$x_i$	$A$	$B$	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	$C$	$P(y_i)$	$\mathcal{I}(y_i)$	$x_i$	$A$	$C$	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	$D$	$P(y_i)$	$\mathcal{I}(y_i)$		
$x_0$	0	0	1/4	0.50		1			$x_0$	0	1	2/4	0.5		1				
$x_1$	0	1	1/4	0.50	$y_0$	1	3/4	0.31		0	1			$y_0$	1	3/4	0.31		
$x_2$	1	0	1/4	0.50		1			$x_1$	1	0	1/4	0.5		1				
$x_3$	1	1	1/4	0.50	$y_1$	0	1/4	0.50	$x_2$	1	1	1/4	0.5	$y_1$	0	1/4	0.5		
$\mathcal{H}(X)$				2.00	$\mathcal{H}(Y)$				0.81	$\mathcal{H}(X)$				1.5	$\mathcal{H}(Y)$				0.81

(c) Gate 3.								(d) Gate 4.											
Input				Output				Input				Output							
$x_i$	$C$	$B$	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	$E$	$P(y_i)$	$\mathcal{I}(y_i)$	$x_i$	$D$	$E$	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	$S$	$P(y_i)$	$\mathcal{I}(y_i)$		
$x_0$	1	1	1/4	0.50	$y_0$	0	1/4	0.50	$x_0$	1	1	2/4	0.5	$y_0$	0	2/4	0.5		
	1	0				1			$x_0$	1	1			$y_0$	0				
$x_1$	1	0	2/4	0.50	$y_1$	1	3/4	0.31	$x_1$	0	1	1/4	0.5	$y_1$	1	2/4	0.5		
$x_2$	1	1	1/4	0.50		1			$x_2$	1	0	1/4	0.5		1				
$\mathcal{H}(X)$				1.50	$\mathcal{H}(Y)$				0.81	$\mathcal{H}(X)$				1.5	$\mathcal{H}(Y)$				1

(e) Gate 5.									
Input				Output					
$x_i$	$E$	$B$	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	$C_{out}P(y_i)$	$\mathcal{I}(y_i)$		
	1	0	2/4	0.5		0			
$x_0$	1	0			$y_0$	0	3/4	0.31	
$x_1$	0	1	1/4	0.5		0			
$x_2$	1	1	1/4	0.5	$y_1$	1	1/4	0.5	
$\mathcal{H}(X)$				1.50	$\mathcal{H}(Y)$				0.81

Note that the exact evaluation of Landauer's limit requires a calculation for each gate and for all possible primary inputs combinations. For example, in the case of the half-adder circuit, which has 5 gates and 2 input bits, it requires  $5 \text{ gates} \times 2^2 \text{ combinations} = 20$  calculations. Hence, the runtime of this method grows exponentially with the number of primary inputs and linearly with the number of gates.

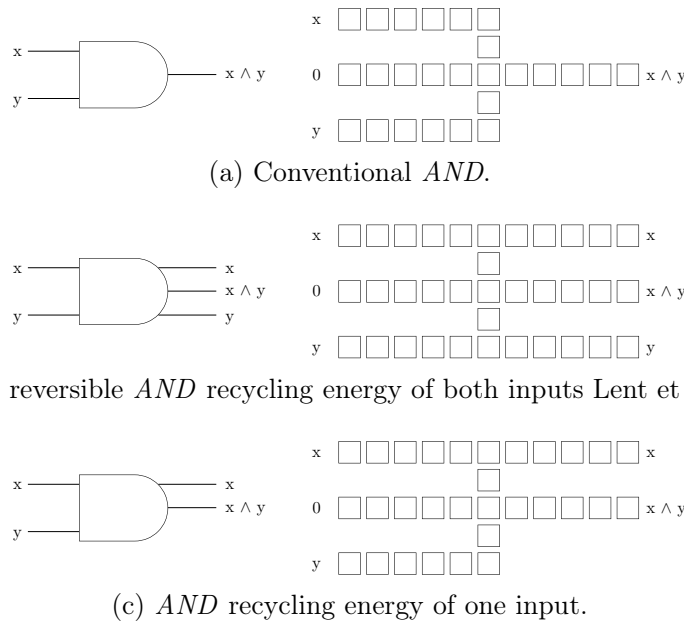


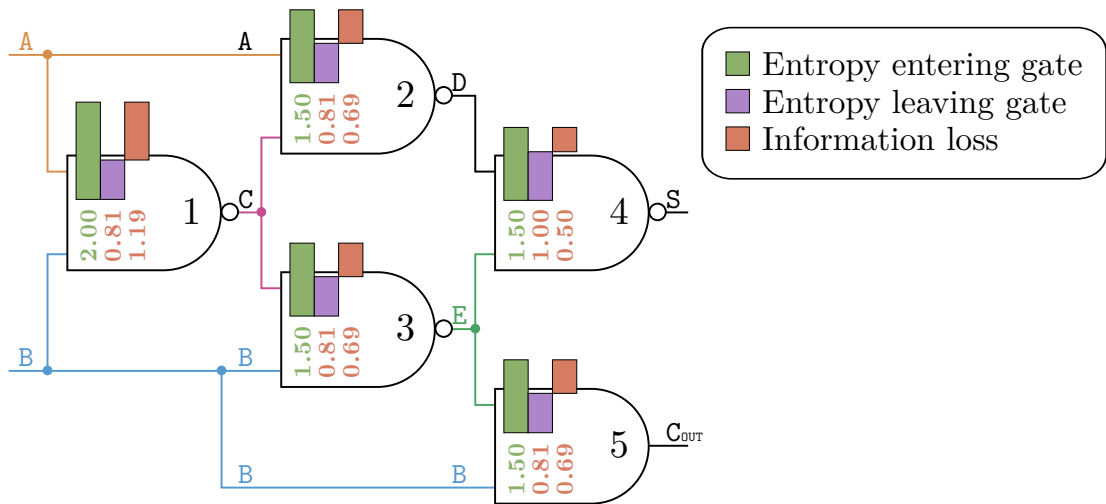
Figure 4.2: Symbols and layouts of conventional as well as  $n$ -bits recycling QCA *AND* gates.

## 4.2 Energy Reduction by Layout Changes

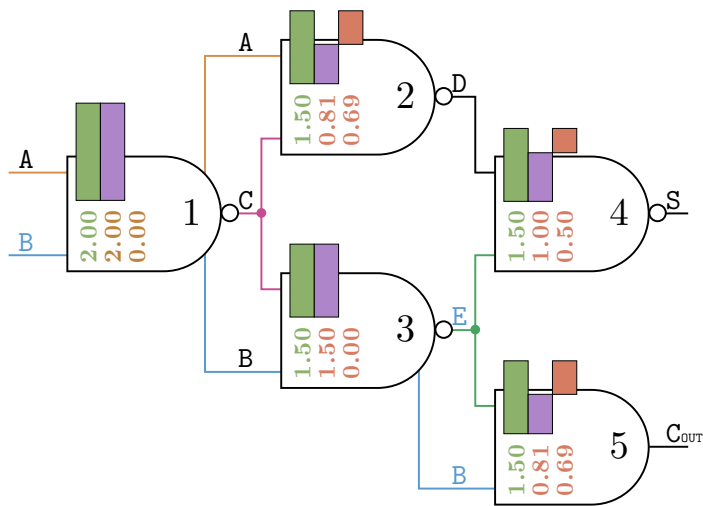
Information losses quantified by the Landauer’s limit are the result of the local states’ merging that happens within conventional logic gates, e. g., *AND*, *OR*, or *MAJ*. Thus, these losses directly depend on the probability distribution of the initial gate states. Such gate can execute a reversible operation and consequently could spend less than  $k_B T \ln(2)$  Joules when the probability distribution of initial states allows a 1-to-1 relation with the probability distribution of the final states [DeBenedictis et al., 2016].

Consider Fig. 4.2 which depicts three different implementations of a QCA *AND* gate: Fig. 4.2a conventional, Fig. 4.2b unconditionally reversible (2-bit recycling gate) proposed by Lent et al. [2006] and Fig. 4.2c conditionally reversible that recovers information of one input (1-bit recycling gate) proposed in this work. Lent showed that the conventional QCA *AND* (Fig. 4.2a) must dissipate energy above  $k_B T \ln(2)$  joules, when the value of one input differs from the output. They also demonstrated that the dissipation of the fully reversible (2-bit recycling) QCA *AND* (Fig. 4.2b), which always guarantees a 1-to-1 relation between its initial and final states, can remain below  $k_B T \ln(2)$  Joules.

We propose here a generalization of the proposal of Lent by echoing only a subgroup of inputs to the output, e. g., the QCA *AND* gate depicted in Fig. 4.2c. That means, these modified gates only operate reversibly for a subset of the  $2^n$  possible



(a) Original half-adder.



(b) Half-adder circuit that exploits fan-outs without degrading delay.

Figure 4.3: Application of proposed method for the half-adder circuit.

Table 4.2: Entropy calculations for gates in the half-adder circuit depicted in Fig. 4.3b

(a) New gate 1 with embedded signals A and B.

Input					Output						
$x_i$	A	B	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	A	C	B	$P(y_i)$	$\mathcal{I}(y_i)$	
$x_0$	0	0	1/4	0.5	$y_0$	0	1	0	1/4	0.5	
$x_1$	0	1	1/4	0.5	$y_1$	0	1	1	1/4	0.5	
$x_2$	1	0	1/4	0.5	$y_2$	1	1	0	1/4	0.5	
$x_3$	1	1	1/4	0.5	$y_3$	1	0	1	1/4	0.5	
$\mathcal{H}(X)$					2	$\mathcal{H}(Y)$					2

(b) New gate 3 with embedded signal B.

Input					Output						
$x_i$	C	B	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	E	B	$P(y_i)$	$\mathcal{I}(y_i)$		
$x_0$	0	1	1/4	0.5	$y_0$	1	1	1/4	0.5		
$x_1$	1	0	2/4	0.5	$y_1$	1	0	2/4	0.5		
	1	0				1	0				
$x_2$	1	1	1/4	0.5	$y_2$	0	1	1/4	0.5		
$\mathcal{H}(X)$					1.5	$\mathcal{H}(Y)$					1.5

initial states, i. e., the ones that have a 1-to-1 relation with the final state. Moreover, *AND* and *OR* gates are indeed *MAJ* gates with one input fixed. Therefore, the same layout also allows the recovery of 2 out of 3 bits on *MAJ* gates. Finally, the possibility of recovering 3 bits on *MAJ* gates remains for future investigation.

Note that logic gates with a 1-to-1 relation between inputs' and outputs' states only guarantee the reversibility at the gate level, but this is not sufficient to enable reversibility at the circuit level. Data flow without discarding information between gates is also needed to accomplish arbitrarily small energy levels. Hence, we propose the exploitation of signals that are used more than once in the circuit. This gives us the ability to add gates that recycle input bits such that we can reduce the fundamental energy limit. Following example shall detail this approach.

**Example 2** *Fig. 4.3a shows the half-adder circuit with its internal fan-outs A, B, C, and E highlighted in orange, blue, pink, and green, respectively. Note that this circuit solely applies conventional gates. As shown in Example 1 (Tables 4.1a to 4.1e), it dissipates 3.76 bits following Ercan's method.*

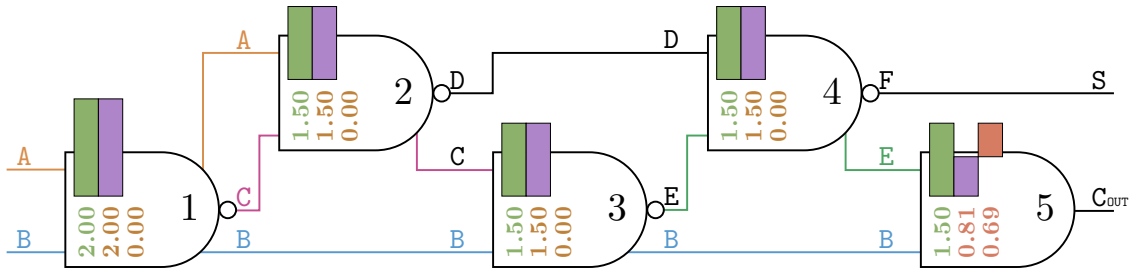


Figure 4.4: Half-adder circuit that exploits fan-outs with delay penalty.

Looking at the circuit, one can notice that all highlighted signals feed more than one gate. Hence, instead of using a fan-out structure that multiplies the signal, one can feed it into gates that recycle its input energy and propagate the signal throughout them to the following stages. The resulting circuit is shown in Fig. 4.3b. Here, gate 1 has been exchanged by its 2-bit recycling version that passes its input A and B to its output (as in Fig. 4.2b). Gate 3 also could be replaced by a 1-bit recycling version that passes its input B to its output (as in Fig. 4.2c). Tables 4.2a and 4.2b shows the new calculation for gates 1 and 3. Landauer’s limit evaluation for this circuit results in a loss of 1.88 bits. The delay of this circuit is identical to the initial version.

If the delay is of lower priority, one can exploit that still some signals are inputs to more than one gate, e. g., signals C and E. Thus, instead of processing a signal in parallel, one can serialize the access and apply it as an echoed input to gates that recycle inputs. Fig. 4.4 depicts the resulting circuit, in which signal F also passes through the 1-bit recycling gate 2 before entering gate 3. Also, signal E traverses gate 4 before being delivered to gate 5. Tables 4.3a and 4.3b show the new calculation for those gates. This additional modification comes at the costs of an increased delay from 3 to 4 stages. However, this reduces the fundamental energy loss down to 0.69 bits.

We carried out an analysis with QCADesigner-E and contrasted the returned energy with the fundamental limits. First, we produced the QCA layouts for the three half-adder versions, i.e., the original netlist, the depth-oriented, and the energy-oriented versions (all showed in Figure 4.5). All designs have the same number of QCA cells. For each of these circuit’s layouts, we did simulations for all input combinations. Table 4.4 shows the averaged energy values measured by QCADesigner-E using an uniform distribution and the energy fundamental limits. Table 4.5 shows the simulation and technology parameters.

The original half-adder layout presents a loss of  $19.50 k_B$  Joules. In this case, the fundamental energy limit represents 19% of the total energy returned by QCADesigner-E. In the depth-oriented and energy-oriented layouts, the energy fundamental limit is

Table 4.3: Entropy calculations for gates in the half-adder circuit depicted in Fig. 4.4

(a) New gate 2 with embedded signal D.

Input					Output				
$x_i$	A	C	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	D	C	$P(y_i)$	$\mathcal{I}(y_i)$
$x_0$	0	1	2/4	0.5	$y_0$	1	1	2/4	0.5
	0	1				1	1		
$x_1$	1	0	1/4	0.5	$y_1$	1	0	1/4	0.5
$x_2$	1	1	1/4	0.5	$y_2$	0	1	1/4	0.5
$\mathcal{H}(X)$				1.5	$\mathcal{H}(Y)$				1.5

(b) New gate 4 with embedded signal E.

Input					Output				
$x_i$	D	E	$P(x_i)$	$\mathcal{I}(x_i)$	$y_i$	F	E	$P(y_i)$	$\mathcal{I}(y_i)$
$x_0$	1	1	2/4	0.5	$y_0$	0	1	2/4	0.5
	1	1				0	1		
$x_1$	0	1	1/4	0.5	$y_1$	1	1	1/4	0.5
$x_2$	1	0	1/4	0.5	$y_2$	1	0	1/4	0.5
$\mathcal{H}(X)$				1.5	$\mathcal{H}(Y)$				1.5

Table 4.4: Energy values in  $k_B$  Joules for half-adder circuits.

	QCADesigner-E	Fundamental Limit
Original	19.50	3.76
Depth-oriented	15.10	1.88
Energy-oriented	8.49	0.69

12% and 8% of the total energy dissipation. One can see that recycling bits indeed reduces energy. Under lower frequencies, the fundamental energy took a more substantial amount of total energy [Torres et al., 2019]. It occurs due to the reduction of friction-like losses.

### 4.3 The Bit-Recycling Algorithm

Algorithm 4 implements the proposed method for synthesized circuits given as netlist. It starts by iterating over all fan-outs (represented by nodes that distribute an output to different targets) in a reverse topological order so that nodes to be processed are not

Table 4.5: Technology and simulation parameters in the tool *QCADesigner-E*

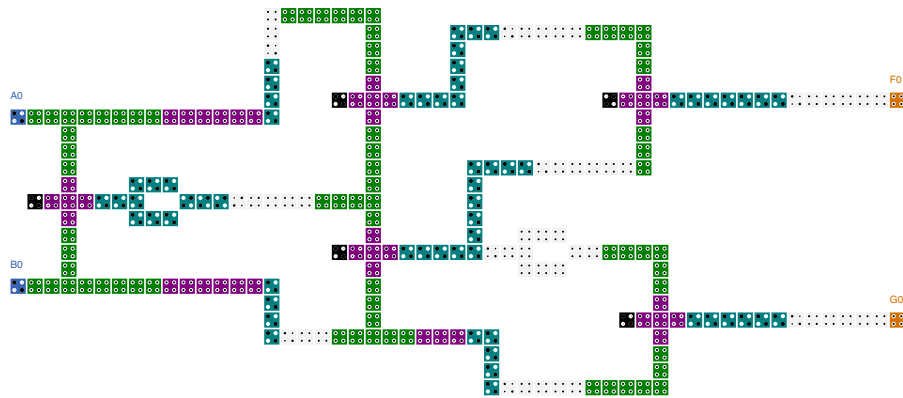
Parameter	Description	Standard Value
QD size	Size of a quantum dot	5 nm
Cell area	Dimensions of each cell	18 nm x 18 nm
Cell distance	Distance between two cells	20 nm
$\tau$	Relaxation time	1E-15 s
$\gamma_H$	Max. saturation energy of clock signal	9.8E-22 J
$\gamma_L$	Min. saturation energy of clock signal	3.8E-23 J
$\epsilon_r$	Relative permittivity of material for QCA system	12.9*
Temp	Operating temperature	1 K
$r_{\text{effect}}$	Maximum distance between cells whose interaction is considered	80 nm <sup>†</sup>
$T_\gamma$	Period of the clock signal	10E-12 s
$\gamma_{\text{slope}}$	Rise and fall time of the clock signal slopes	1E-10 s
$\gamma_{\text{shape}}$	Shape of clock signal slopes [RAMP/GAUSSIAN]	GAUSSIAN
$T_{in}$	Period of the input signals	10E-12 s
$T_{sim}$	Total simulation time	80E-12 s
$T_{step}$	Time interval of each iteration step	1E-16 s

\* Relative permittivity of GaAs and AlGaAs

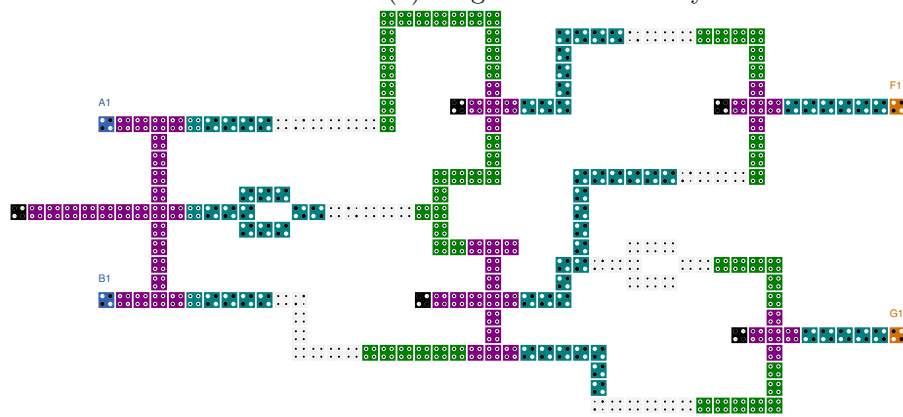
<sup>†</sup> Interaction effects between two cells decays inversely with the fifth power of its distance

affected by a parent’s recycling (Line 1). For each of these nodes, the algorithm builds chains of gates by selecting unique children from their outputs (Lines 2 to 8). The rank provided by the function `topological-order`, i. e., the node’s distance from the primary inputs, is also fundamental for building those chains. The algorithm first iterates on a sorted-by-rank list containing sets of children nodes. Each set contains the children that are on a same rank, created by the function `ranked-children` (Line 4). Thus, each one of these sets is fed to a function `choose` which evaluates its nodes and selects a valid one that can be recycled and added to the chain (Line 5). Besides discarding invalid nodes, this function also enables the definition of gate’s priorities, e. g., 1-bit recycling reversible gates over conventional ones. If energy shall be prioritized, it may even select multiple gates within same set (It will be clear in the following examples). In this work, we employed a straightforward `choose` that only selects the first valid nodes. The selected nodes are added to the `choices` list (Line 6) and removed from the set (Line 7). Finally, when the chain is complete, i. e., there are no sets left, it selects a

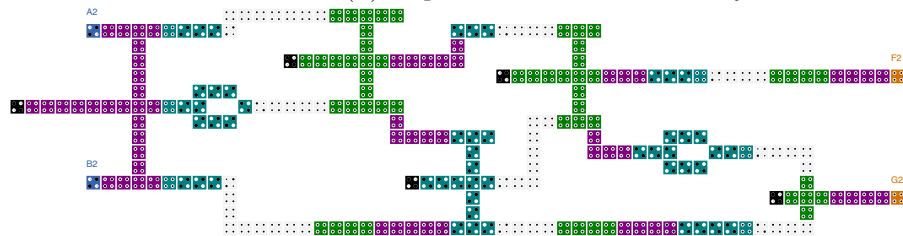




(a) Original half-adder layout.



(b) Depth-oriented half-adder layout.



(c) Energy-oriented half-adder layout.

Figure 4.5: Half-adder circuit QCA layouts.

non-recyclable node, e. g., an output, to be at the bottom of the chain. After selection of the children, the algorithm links their inputs and outputs together (Line 8). This is done by the function `make-chain`. The execution stops when the choices list has at most one node left (Line 9), since at this point it cannot build any other chain from the fan-out.

The examples in Figs. 4.6 and 4.7 demonstrate the operation of the algorithm for two circuits.

**Example 3** *Fig. 4.6 shows the generation of chains in the depth-oriented strategy. In*

---

**Algorithm 4:** Building chains of  $n$ -bits recycling gates.
 

---

```

data :  $netlist \leftarrow$  circuit's netlist
1 foreach  $node \in reverse\text{-topological}\text{-order}(netlist)$  do
2   repeat
3      $choices \leftarrow \emptyset$ 
4     foreach  $set \in ranked\text{-children}(node)$  do
5        $sel\text{-children} \leftarrow choose(set)$ 
6        $choices \leftarrow choices \cup sel\text{-children}$ 
7        $set \leftarrow set \setminus sel\text{-children}$ 
8     make-chain}(node, choices)
9   until  $|choices| \leq 1$ 

```

---

the initial version of the circuit (Fig. 4.6a) the output of node N1 connects to all other nodes. During the first iteration (Fig. 4.6b), the algorithm selects and merges nodes A1, B1 and O1 into a chain that starts at node N1 (marked as blue lines). Further, nodes A1 and B1 are changed to its 1-bit recycling versions. In the following iteration (Fig. 4.6c), a second chain consisting of nodes A2 and B2 is created (marked as red lines), and node A2 is modified to a 1-bit recycling gate.

Fig. 4.7 shows the operation of the energy-oriented strategy in another circuit. In this case the algorithm only need one iteration to build the chain since it picks all children, despite some of them (A1 and A2) being in the same rank.

Finally, the circuit from Fig. 4.8 illustrates how fully reversible gates are integrated. Initially, Fig. 4.8a shows two nodes, N1 and N2, each one duplicating its outputs and having a common children, A1. In the first iteration (Fig. 4.8b), the connection between nodes N1 and O1 is placed between nodes A1 and O1 (blue lines). Further, node A1 is modified to a 1-bit recycling gate. In the second iteration (Fig. 4.8c), the connection between nodes N2 and O2 is moved between nodes A1 and O2 (red lines). Now, node A1 is changed to a fully reversible gate.

In order to assess our strategies, we evaluated the Landauer's limit for well-established benchmarks. Having in mind that the computational cost grows exponentially with the number of circuit's primary inputs (see also Section 4.1), we analyzed circuits with up to 40 input bits of the MCNC (154 circuits) and the EPFL (6 circuits) [Yang, 1991; Amarù et al., 2015]. Table 4.6 presents the obtained results for 14 circuits of MCNC and 6 majority-based circuits produced by Testa based on the EPFL benchmarks specifications [Testa et al., 2017]. All benchmarks were evaluated *as they were*, i. e., we did not modify them before employing our technique.

Table 4.6 lists the experiments results. Here, the first column shows the bench-

Table 4.6: Benchmark results

Benchmark	Original			Depth-Oriented			Energy-Oriented			
	Size	I/O	Energy	Delay	Energy	Reduction	Energy	Reduction	Delay	Increase
<i>log2</i>	37584	32/32	29681.76	181	22546.76	24%	9377.48	68%	1877	10.37 ×
<i>sin</i>	7895	24/25	6206.10	91	4642.82	25%	1950.14	69%	762	8.37 ×
<i>caclc</i>	745	10/11	781.34	11	511.96	34%	259.98	67%	114	10.36 ×
<i>dec</i>	420	8/256	412.72	4	412.72	0%	25.02	94%	290	72.50 ×
<i>int2float</i>	246	11/7	259.10	9	188.88	27%	88.55	66%	52	5.78 ×
<i>ctrl</i>	127	7/26	135.60	5	92.31	32%	38.42	72%	36	7.20 ×
<i>prom1</i>	7803	9/40	6862.44	24	3560.18	48%	1535.11	78%	869	36.21 ×
<i>mainpla</i>	5346	26/54	4686.49	38	2211.91	53%	1326.54	72%	583	15.34 ×
<i>xparc</i>	5105	39/73	4642.60	47	1742.30	62%	915.88	80%	481	10.23 ×
<i>bca</i>	3669	16/46	3458.48	26	1334.85	61%	546.24	84%	402	15.46 ×
<i>prom2</i>	3513	9/21	3115.70	22	1696.55	46%	731.23	77%	481	21.86 ×
<i>apex4</i>	3452	9/19	3062.46	21	1645.90	46%	825.64	73%	404	19.24 ×
<i>ex1010</i>	3340	10/10	3038.40	24	1642.23	46%	718.00	76%	392	16.33 ×
<i>bcb</i>	3314	16/39	3117.99	26	1247.13	60%	518.97	83%	354	13.62 ×
<i>bcc</i>	3238	16/45	3073.95	25	1196.35	61%	503.59	84%	353	14.12 ×
<i>C6288</i>	2337	32/32	1665.15	120	1090.78	34%	350.66	79%	208	1.73 ×
<i>bcd</i>	2304	16/38	2182.07	25	934.01	57%	391.27	82%	243	9.72 ×
<i>table3</i>	2183	14/14	2009.57	24	950.47	53%	432.81	78%	258	10.75 ×
<i>table5</i>	1987	17/15	1841.59	26	841.46	54%	390.88	79%	211	8.12 ×
<i>cps</i>	1928	24/109	1883.59	31	812.68	57%	430.58	77%	226	7.29 ×
<b>Average</b>	<b>4827</b>	<b>17/46</b>	<b>4105.86</b>	<b>39</b>	<b>2465.11</b>	<b>44%</b>	<b>1067.85</b>	<b>77%</b>	<b>430</b>	<b>15.73 ×</b>

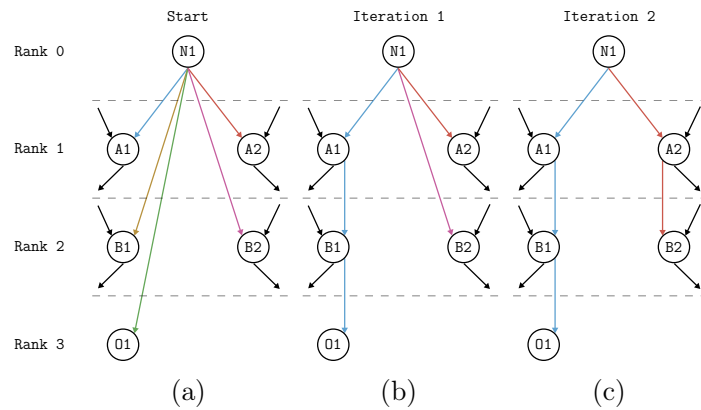


Figure 4.6: Generations of chains in the depth-oriented strategy.

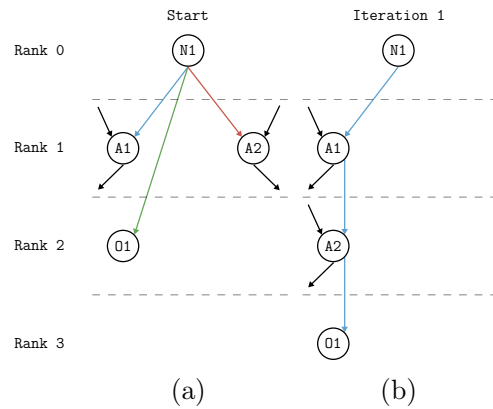


Figure 4.7: Generations of chains in the energy-oriented strategy.

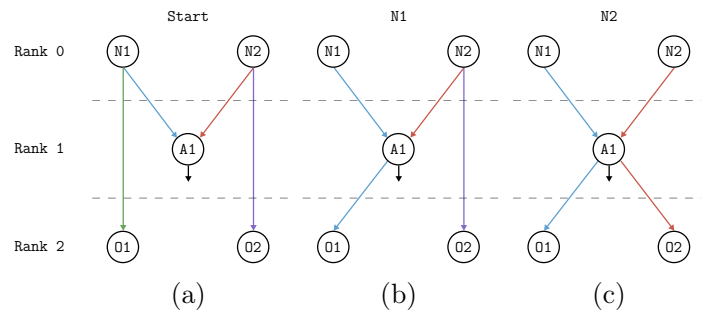


Figure 4.8: Integration of fully reversible gates.

mark names. The second, third and fourth columns display the size of the circuit (number of gates) and the number of inputs and outputs, respectively. The fifth and sixth columns lists the exact Landauer’s limit for an uniform input distribution applied to the original circuits, and its delay (number of levels). The next two columns refer to the depth-oriented results. That means, the seventh column shows the fundamental energy limit, while column eight lists the energy reduction resulting from

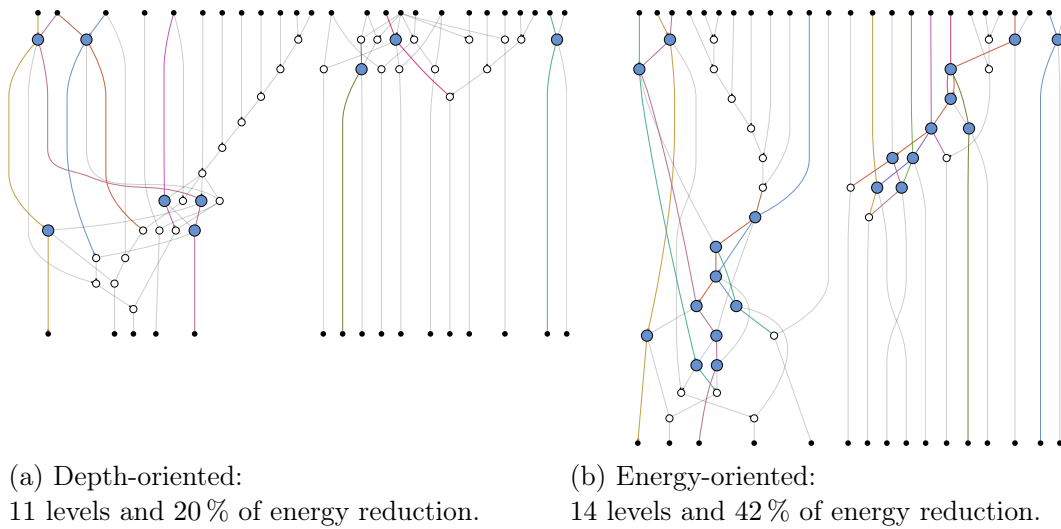
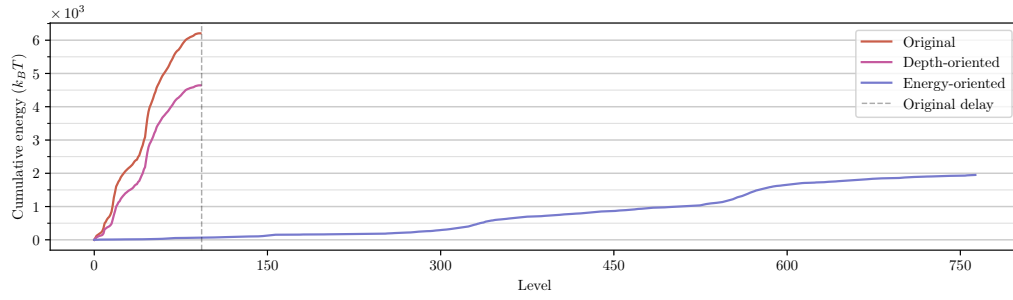


Figure 4.9: Circuit graphs resulting from the proposed strategies: the *i1* circuit's case.

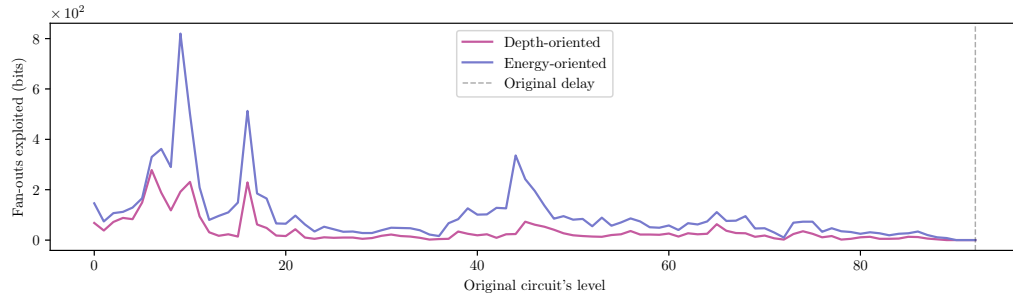
our method. The remaining four columns relate to the energy-oriented results and list the Landauer's limit and its reduction as well as the new delay and its increase. We count the delay as the logic depth on the circuits' critical paths. Therefore, we are considering the best (minimum) possible delays. Note that both strategies come at no costs in terms of gate area. Hence, the circuit size is the same for the original and both modified cases.

The simulation results reveal that the fundamental energy limits could be reduced by, on average, 44 % (up to 62 %) on the depth-oriented case. If the delay was not restricted, i. e., the energy-oriented case, fundamental energy limits decreased by, on average 77 % (up to 94 %) at the costs of an average maximum delay increase of  $16\times$  (up to  $73\times$ ).

For the sake of clarity, we choose a small circuit, not presented in Table 4.6, to show the new logic networks produced by the proposed strategies. Fig. 4.9 shows the circuit produced by the proposed algorithms for the circuit *i1* (MCNC benchmark). Each color in the edges represents a signal for a specific fan-out, while the blue nodes highlight bit recycling gates. Black nodes represent the circuit's primary inputs and outputs. Fig. 4.9a displays the depth-oriented result for this circuit where 9 conventional gates were replaced by its recycling counterparts. This results into reduction of the fundamental energy dissipation by 20 % without any delay change. Fig. 4.9b depicts for the same circuit the outcome of the energy-oriented algorithm. Here, the fundamental energy dissipation can be reduced by 42 % due to the serialization of some information transfer. Note that both graphs are aligned level by level allowing the comparison of the delay degradation (from 11 to 14 levels). Note that we do not consider



(a) Exact cumulative energy loss.



(b) Fan-outs exploited per level.

Figure 4.10: *sin* circuit's evaluation.

input and output levels.

Fig. 4.10 presents the loss rate pattern and the fan-outs density for the EPFL circuit *sin*. First, Fig. 4.10a shows the accumulated energy losses for the original circuit version, the new depth-oriented version, which has the same delay as the original (vertical dashed line), and the energy-oriented version. Then, Fig. 4.10b presents the density fan-outs exploited by the depth-oriented and energy-oriented techniques.

It is interesting to note that circuits profit differently from both strategies, i. e., the ranking of the improvement of the energy limit for depth-oriented and energy-oriented results is not the same. This follows from the fact that the first approach takes advantage from the number of fan-outs that target different levels while the latter benefits only from the number of fan-outs. The *dec* benchmark is a good example for this observation. It has a high number of fan-outs that possess many targets on the same level. This explains the negligible improvement of the energy limits when delay is not impaired and the high gain when there is no such restriction.

## 4.4 Summary

In this chapter, we proposed a step towards feasible partially reversible circuits designed for Field-Coupled Nanocomputing technologies that enable considerable reduction of

its fundamental energy limits. The proposed technique identifies exploitable fan-outs that can be embedded in logic gates to reduce local information losses without any delay penalties. When delay is not a concern, energy can be decreased even more by serializing gates and exploiting more fan-outs.

# Chapter 5

## Towards a Unified Analysis

Chapters 3 and 4 presented energy reduction techniques based on clock and layout manipulation, respectively. Considering the proposed techniques, it is reasonable to expect a unified analysis with both methods. In this chapter, we introduce several perspectives that can complete this investigation. In Section 5.1, we explain the differences in energy evaluation in our techniques. Next, we carried out two orthogonal analyses. In the first study, presented in Section 5.2, we discuss how layout-based methods restrict clock-based solutions. For that, we introduced a new configuration space for circuits (an Energy-Throughput-Depth Space). Finally, in Section 5.3, we analyze how our techniques benefit from different initial netlists for the same specification. More precisely, we contrast our energy reduction methods applied in both the EPFL initial netlists used in Chapter 3 and Testa’s depth-optimal netlists used in Chapter 4. These perspectives open relevant opportunities for future works.

### 5.1 Energy Evaluation

The attentive reader may have already realized that we used different methods to evaluate the energy in each case. Strange as it may seem, there is a reasonable explanation for this difference. In a partially reversible system based on layout changes (Landauer clocked circuits), the energy evaluation relies on Shannon’s entropies of input and output probability distributions. Specifically, these losses happen because typically these gates have fewer degrees of freedom in their outputs than they have in their inputs. As information and erasure waves flow only in one direction in these systems, i.e., from inputs to outputs, the gate can only transfer to the output at most the amount of information that fits the output’s degrees of freedom. Therefore, if there is any other part of the original information missing from the outputs, then it is necessarily ejected



in the form of heat to the environment. Information loss, in this case, is usually a fractional number. This evaluation method follows Landauer’s original work and Ercan’s approach for QCA systems [Landauer, 1961; Ercan and Anderson, 2013].

Unfortunately, the evaluation of Shannon’s entropy for all gates in a circuit comes at a high computational cost with exponential time and space complexity, rendering it laborious for comprehensive designs. Thus, when the available resources are not enough to compute the circuit’s energy limit, we adopt a more straightforward approach. The main idea is the summation of the upper loss for each gate in the design, i.e., the maximum entropy that a gate can receive. Note that by this choice, we are overestimating losses since we are not considering conditionally reversible cases. In the case of conventional gates, e.g., AND, OR, NAND, MAJ, we count the number of variable input bits as losses. In other cases, we consider each embedded fan-out as a recycled bit, and they are not considered losses.

In the partially reversible pipelines, the losses always happen in input wires for each stage. Specifically, the erasure occurs in a place without any other adjacent part that still preserves the information. Therefore, this loss is inevitable and, even if the bit has different probabilities for 0 and 1 values, the loss is at most 1 bit of entropy. The information loss is valued as an integer number (the sum of input wires for each stage) even though the wires do not carry an integer amount of information (Shannon’s Entropy). This evaluation method derives from some of our primary supporting references [Frost-Murphy, 2009; Ottavi et al., 2011; Ercan and Anderson, 2011; Stearns and Anderson, 2013].

## 5.2 Energy-Throughput-Depth Spaces

The analysis of partially reversible pipelines revealed the energy-throughput relations for these systems, as showed in Chapter 3. Typically, as we improve throughput, energy deteriorates and vice-versa. The viable partially reversible pipeline configurations lost less energy than the Landauer clocked configuration did, i.e., the latter determines a frontier of the energy-throughput configuration space (best throughput). The pure Bennett clocked configuration defines the other border (best energy). Based on that relation, we presented the OSA algorithm that produces the optimal solution for this energy-throughput tradeoff. Our algorithm returns the best (smallest) energy for the minimum tolerable throughput. Moreover, our algorithm can yield the best (biggest) throughput for the maximum acceptable energy loss. With both values, we explored a granular space of viable energy-throughput solutions (a Pareto frontier).

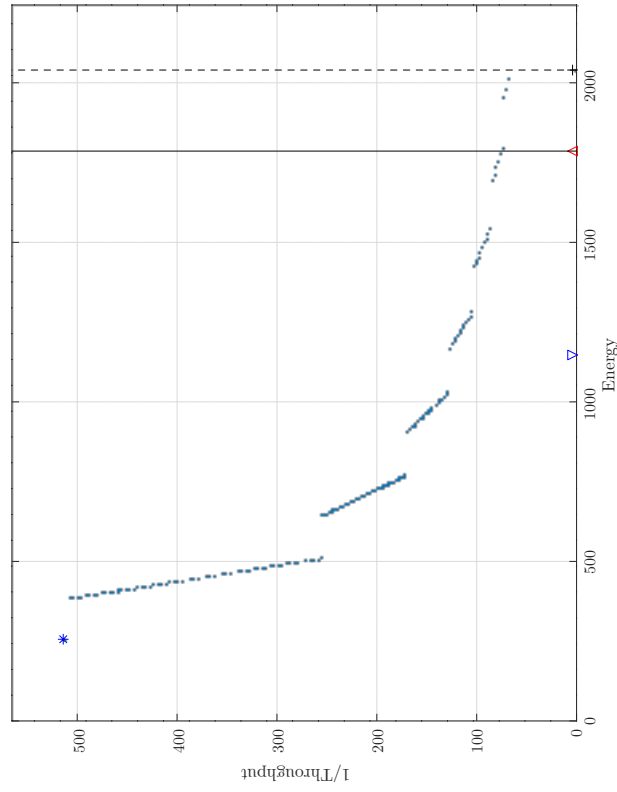
Additionally, throughput and depth are linked in partially reversible pipelines. As shown in Chapter 3, our optimization algorithm maximizes throughput by minimizing the maximum stage size,  $m$ . The throughput is a function of  $m$  ( $1/2m+2$ ), and the depth is equal to the maximum stage size times the number of stages, i.e., each stage must have the same delay as in a normal pipeline.

The partially reversible FCN circuits based on layout changes (Landauer clocked circuits), on the other hand, do not degrade the throughput at all. This system either enhances energy without depth degradation (depth-oriented) or improves energy even more at the cost of depth deterioration, as revealed in Chapter 4. These energy reduction strategies are implemented by a heuristic algorithm, i.e., returned energy and depth are probably not the best solutions. The optimal solution and a granular exploration for this energy-depth space (analogous to our energy-throughput relation for partially reversible pipelines) are an open problem under investigation.

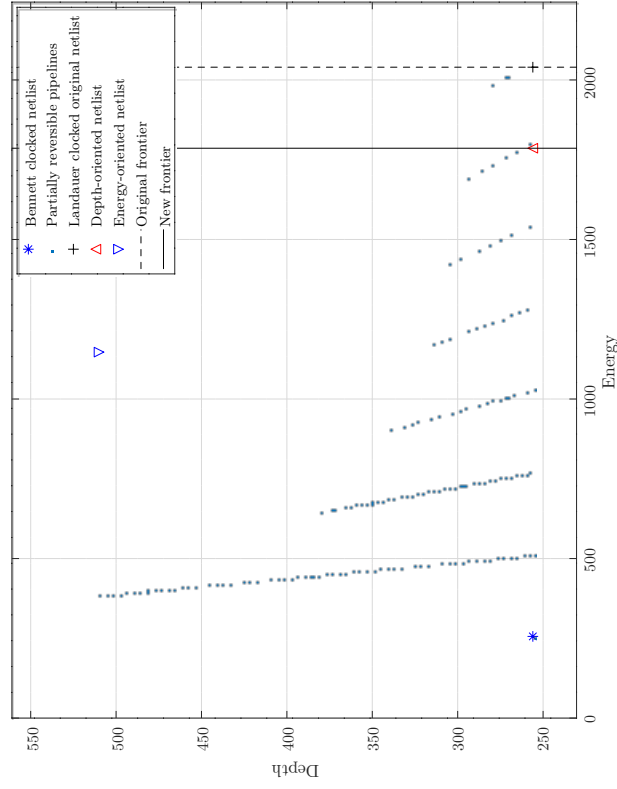
To analyze both sorts of systems, we need to merge those two-dimensional problems in one three-dimensional space, i.e., an Energy-Throughput-Depth space. We also applied the Pareto dominance to the data analyzed in this Chapter, as in Chapter 3, i.e., we are only considering configurations in the Pareto set. Let us take an example of configuration space from the Adder benchmark case (EPFL’s initial netlist) presented in Figure 5.1. For the sake of clarity, we present two perspectives of the configuration space bounded by the EPFL’s Landauer clocked netlist. As depth-oriented partially reversible circuits improve energy without any throughput or depth degradation, this configuration defines a new frontier for viable partially reversible pipelines. This edge also reduces the number of feasible pipeline configurations.

Table 5.1 shows some information about the configuration spaces for all other EPFL benchmarks. The first column shows the benchmark name, the second to fifth columns present data about EPFL’s initial netlists. Specifically, the second column shows the netlist’s original depth. The third, fourth, and fifth columns display three sets with the number of viable pipelines bounded by Landauer clocked circuits (original, depth-oriented, and energy-oriented). Specifically, the third column presents the Set Original, which is the Pareto set yield from all pipelines returned by the OSA algorithm and the original circuit (Landauer-clocked). Set Depth, the Pareto set yield from the Set Original and the depth-oriented netlist, is presented in the fourth column. Finally, Set Energy, shown in the fifth column, is the Pareto set yield from the Set Depth and the energy-oriented netlist. The sixth to ninth columns show the analogous information for Testa’s depth-optimal EPFL netlists.

As EPFL’s initial netlists are always deeper than Testa’s (comparing Levels columns), the former invariably has more viable pipelines than the latter. Let us



(a) Throughput-Energy perspective.



(b) Depth-Energy perspective.

Figure 5.1: Configuration space (Throughput-Energy-Depth) for EPFL's Adder.

Table 5.1: Configuration spaces

Benchmark Name	EPFL's netlists				Testa's netlists				
	Levels	Number of configurations		Levels	Number of configurations		Levels	Number of configurations	
		Set Original	Set Depth		Set Energy	Set Original		Set Depth	Set Energy
Adder	256	160	156	157	14	13	13	14	14
Round-robin arbiter	88	22	20	21	16	13	14	14	14
Coding-cavlc	17	16	16	17	13	12	12	13	13
Alu control unit	11	10	9	10	7	6	6	7	7
i2c controller	21	19	19	20	11	9	10	10	10
Int to float converter	17	16	16	17	11	10	10	11	11
log2	445	245	244	245	183	115	116	116	116
Max	288	235	230	231	29	23	24	24	24
Memory controller	115	98	97	98	20	15	15	16	16
Multiplier	275	216	215	216	113	78	78	79	79
Priority encoder	251	150	145	146	12	11	11	12	12
Lookahead XY router	55	45	44	45	11	9	9	10	10
Sine	226	141	140	141	93	59	59	60	60
Square-root	5059	916	910	911	692	216	217	217	217
square	251	187	186	187	38	32	32	33	33
voter	71	57	57	58	62	54	54	55	55

retake the Adder circuit as an example with its data from Table 5.1. Considering the EPFL netlist, the Set Original has 160 different viable pipelines. Now, considering the depth-oriented netlist, the Pareto set shrinks to 156 feasible pipeline configurations (Set Depth). That reduction occurs because the depth-oriented netlist has the same throughput and depth as the original circuit (Landauer-clocked) but it requires less energy, eliminating four pipelines from the set (also shown in Figure 5.1). The Set Energy, though, does not shrink the space of viable configurations, but it adds energy-oriented configuration.

A similar pattern occurs when considering Testa’s netlist, but with less diversity. Set Depth is in most cases equal to Set Original, in the number of viable configurations (the original circuit is replaced by the depth-oriented one). Also, Set Energy only adds one more configuration (the energy-oriented).

### 5.3 Energy and Synthesis’ Depth Reduction

When it comes to designing a logic network, different goals might be pursued. For instance, the designer might expect to minimize the size of the network, i.e., the number of logic gates. Rather, the designer might want to reduce the depth of the circuit, i.e., the critical path length. These different goals for synthesis algorithms differently impact our energy reduction techniques. In this section, we analyze the influence of reducing depth in energy reduction opportunities. Specifically, we contrast size, energy, depth, the number of fan-outs, and throughput from the EPFL initial netlists against Testa’s depth optimal netlists.

First, we focus on Landauer clocked circuits. Table 5.2 presents our results, which are all ratios of Testa’s depth optimal netlists over EPFL initial netlists. The first column shows benchmark names. The second, third, and fourth columns display data related to original netlists, specifically the size of the circuit (number of gates) and approximate energy and depth, respectively. The next two columns refer to results from depth-oriented strategy. That means the fifth column shows the number of exploitable fan-outs, while column six lists the approximate energy limit resulting from our method. The remaining three columns relate to the results from energy-oriented strategy. They register the number of exploitable fan-outs, the approximate energy, and the new depth.

Looking only at both EPFL initial and Testa’s original netlists, we can arrange EPFL circuits in three groups. The first collection, almost entirely composed of arithmetic circuits, includes the Adder, the coding-cavlc, the log2, the max, the multiplier, the sine, the square-root, the square, and the voter. In this group, the improvement

in depth (Testa’s netlist) is related to a degradation in size and energy. When our depth-oriented technique is applied to these circuits, we can see that Testa’s netlists have more exploitable fan-outs than EPFL initial netlists do; hence they have more energy reduction opportunities. Even with more opportunities, the number of fan-outs was not enough to compensate for the increase in size and spent energy. This pattern remains even when we apply our energy-oriented technique.

The second group, entirely composed of random/control benchmarks, contains the round-robin arbiter, the ALU control unit, the i2c controller, the memory controller, the priority encoder, and the router. Depth reduction results in an improvement in all other metrics, e.g., size, energy, either in original, depth-oriented, or energy-oriented netlists. Finally, to describe the int to float converter, the significant reduction in depth does not result in meaningful changes in size or energy.

Looking at partially reversible pipelines, Table 5.3 presents data about EPFL initial netlists and Testa’s depth optimal netlists. The first column shows the benchmark names. The second, third, and fourth columns display data related to EPFL initial netlists. More accurately, the second column shows the average of pipeline energy degradation related to pure Bennett clocked circuits (energy best value). The average of pipeline throughput degradation related to Landauer’s clocked circuits (throughput best value) is displayed in the third column. The fourth column presents the average quality, which we defined as the inverse of the product of throughput degradation and energy degradation. The next three columns show the analogous information for Testa’s depth-optimal EPFL netlists. The best results are highlighted in bold.

As expected, Testa’s depth-optimal netlists present a better (smaller) throughput degradation than EPFL initial netlists for all circuits. Regarding energy degradation, though, circuits could be grouped in similar classes to the Landauer clocked circuits mentioned above. The arithmetic circuits present a smaller energy degradation for EPFL’s netlist than Testa’s. The second group composed of random/control benchmarks shows smaller energy degradation for Testa’s netlist than EPFL. The quality, on the other hand, is always better for Testa’s netlists.

## 5.4 Summary

Our analyses reveal some essential details. Examining the configuration space for a specific netlist, as shown in Section 5.2, when energy is a critical concern, the preferred choice will be partially reversible pipelines. On the other hand, if delay and throughput also matter, then layout-based partially reversible systems are the solu-

Table 5.2: Depth reduction effects in Landauer clocked circuits.

Benchmark Name	Original			Depth-oriented			Energy-oriented		
	Size	Energy	Depth	Fanouts	Energy	Depth	Fanouts	Energy	Depth
adder	2.92	3.59	0.05	5.98	3.26	0.15	4.17	3.15	0.15
cavlc	1.08	1.17	0.76	1.11	1.20	1.05	1.22	1.12	1.05
log2	1.17	1.23	0.41	0.85	1.38	0.47	1.22	1.24	0.47
max	2.51	2.80	0.10	2.82	2.80	1.67	3.20	2.50	1.67
mult	1.55	1.72	0.41	1.41	1.79	2.09	1.77	1.67	2.09
sin	1.46	1.59	0.41	1.24	1.71	1.00	1.58	1.60	1.00
sqrt	2.13	2.27	0.14	2.53	2.20	0.67	2.23	2.30	0.67
square	1.04	1.07	0.15	0.87	1.11	0.56	1.07	1.08	0.56
voter	1.02	1.07	0.87	1.13	1.06	0.99	1.09	1.05	0.99
arbiter	0.18	0.21	0.18	0.11	0.30	0.92	0.21	0.21	0.92
ctrl	0.73	0.76	0.64	0.65	0.80	0.53	0.80	0.71	0.53
i2c	0.83	0.85	0.52	0.76	0.88	0.82	0.86	0.85	0.82
memctrl	0.22	0.22	0.17	0.15	0.25	0.20	0.23	0.22	0.20
priority	0.51	0.54	0.05	0.23	0.68	0.07	0.51	0.56	0.07
router	0.44	0.47	0.20	0.45	0.47	0.17	0.35	0.54	0.17
int2float	0.95	1.02	0.65	0.83	1.08	0.98	1.07	0.97	0.98

Table 5.3: Depth reduction effects in partially reversible pipelines

Benchmark Name	EPFL's netlists			Testa's netlists		
	Energy degradation	Throughput degradation	Quality	Energy degradation	Throughput degradation	Quality
adder	<b>2.9</b>	64.9	0.0066	6.6	<b>4.5</b>	<b>0.0580</b>
log2	<b>86.5</b>	90.7	0.0007	111.3	<b>39.4</b>	<b>0.0013</b>
max	<b>2.9</b>	78.7	0.0061	4.6	<b>7.9</b>	<b>0.0444</b>
mult	<b>21.7</b>	63.3	0.0028	47.5	<b>24.5</b>	<b>0.0030</b>
sin	<b>28.3</b>	44.9	0.0024	61.4	<b>20.3</b>	<b>0.0030</b>
sqrt	<b>28.6</b>	504.6	0.0003	44.7	<b>91.8</b>	<b>0.0009</b>
square	<b>23.1</b>	54.1	0.0024	50.3	<b>9.9</b>	<b>0.0076</b>
voter	<b>2.3</b>	17.4	0.0451	2.4	<b>15.3</b>	<b>0.0482</b>
arbiter	14.6	16.3	0.0099	<b>4.7</b>	<b>5.3</b>	<b>0.0595</b>
cavlc	19.5	5.2	0.0291	<b>19.2</b>	<b>4.3</b>	<b>0.0456</b>
ctrl	9.1	4.0	0.0472	<b>9.0</b>	<b>2.7</b>	<b>0.0832</b>
i2c	3.8	6.5	0.0579	<b>3.5</b>	<b>4.0</b>	<b>0.0923</b>
int2float	7.5	5.2	0.0528	<b>7.0</b>	<b>3.7</b>	<b>0.0822</b>
memctrl	8.6	29.9	0.0077	<b>4.2</b>	<b>6.3</b>	<b>0.0542</b>
priority	2.8	60.7	0.0093	<b>2.0</b>	<b>4.0</b>	<b>0.1622</b>
router	2.8	16.3	0.0265	<b>1.9</b>	<b>4.0</b>	<b>0.1484</b>



tion. Even though we used an approximate energy evaluation for Landauer clocked circuits, that does not change our qualitative conclusions; it only reduces the number of viable pipelines.

When analyzing different netlists for the same circuit specification, as done in Section 5.3, we showed how each technique benefits from depth reduction in the synthesis process. For Landauer clocked systems, one can see that reducing depth not necessarily results in energy improvement. For partially reversible pipelines, we defined a quality measure. This standard is based on the relevant metrics for those systems, i.e., the best values of energy and throughput, both independent of the netlist. This new metric shows that partially reversible pipelines always benefit from depth reduction.

# Chapter 6

## Conclusions

Researchers must focus on energy scalability to establish emerging technologies. This research aimed to identify effective strategies for energy reduction in FCN circuits. Based on an in-depth analysis of information loss circumstances in FCN circuits, one might conclude that local changes in the circuit netlist can reduce fundamental energy limits without any resynthesis process. Therefore, the results indicate an affirmative answer to the question "Can we take advantage of the current state-of-the-art irreversible logic synthesis algorithms and reduce energy limits in a post-synthesis process?"

This thesis resulted in the following contributions. First, we created a novel type of FCN partially reversible system based on conditionally reversible gates. These systems reduce energy without any throughput degradation, differently from the other existing FCN partially reversible systems (pipelines). This new system should be the first option for designers when all performance metrics equally matter since pipelines excessively damage the throughput.

The second main contribution was an enhancement of partially reversible pipelines. We extended those systems in different ways. First, we developed the means to simulate these pipelines in QCADesigner. Second, we proposed optimal algorithms that improve energy-throughput tradeoff and allow granular exploration of the solution space.

Another significant contribution was the design of a unified analysis of FCN partially reversible systems. The study of both systems reveals the best choice for designers based on their project requirements.

Finally, we delivered a clear and concise review of reversible computing and FCN connections. Our review constitutes a significant contribution, as there are several misconceptions about the application of reversible computing in FCN circuits, especially in designing reversible logic gates based on quantum computing gates such as

Fredkin and Toffoli gates. Our Background chapter only highlights the core theory, e.g., Landauer's, Bennett's, Anderson's, and Frank's works, including FCN basics and its clocking systems, revealing the conditions for information loss.

In this thesis, we approached the design of energy-efficient FCN circuits from a different perspective. Our techniques show immense improvements in those FCN systems and they could be adapted for other technologies. Our work also opens other research paths. Finally, we believe this thesis will act as a significant enabler for FCN consolidation.

## 6.1 Future Work

We give some directions for future research.

- **Union of modern FCN placement-and-routing algorithms with our optimal techniques for designing partially reversible pipelines.** As shown in Chapter 3, the energy-throughput tradeoff was designed considering the number of logic gates in the maximum stage size. Placement-and-routing algorithms also increase circuit depth due to restrictions in clocking distribution schemes. Hence, it would be interesting to study the implications in the energy-throughput compromise when dealing with placement-and-routing solutions to acquire tighter bounds for energy and throughput. Further, multiple-objective algorithms could be designed based on this study.
- **Development of optimal algorithms for designing partially reversible systems (Landauer clocked circuits).** Our algorithm for recycling bits presented in Chapter 4 is not optimal. Despite the notable improvements in circuit energy, it could be better. Furthermore, a similar solution to the partially reversible pipelines is desirable, e.g., an optimal algorithm to return the best energy dissipation given a depth restriction or the best depth given an energy dissipation restriction.
- **Development of strategies for a faster evaluation of energy limits.** As shown in Chapter 4, Landauer's limit assessment is computationally expensive. It is possible to improve efficiency by using different data structures and caching strategies. Moreover, it is possible to improve our approximate evaluation presented in Chapter 6.
- **Development of energy-aware synthesis algorithms based on our techniques.**

# Bibliography

- Amarù, L., Gaillardon, P.-E., and De Micheli, G. (2015). The EPFL Combinational Benchmark Suite. In *Proceedings of the 24th International Workshop on Logic & Synthesis (IWLS)*.
- Anderson, N. G. (2013). Gate abstractions and reversibility: On the logical-physical link. In *Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on*, pages 1059--1062. IEEE.
- Anderson, N. G. (2017). Information as a physical quantity. *Information Sciences*, 415:397--413.
- Anderson, N. G. (2018). Landauer's limit and the physicality of information. *The European Physical Journal B*, 91(7):156.
- Anderson, N. G. and Bhanja, S., editors (2014). *Field-Coupled Nanocomputing: Paradigms, Progress, and Perspectives*. Springer Berlin Heidelberg.
- Ardesi, Y., Wang, R., Turvani, G., Piccinini, G., and Graziano, M. (2019). Scerpa: a self-consistent algorithm for the evaluation of the information propagation in molecular field-coupled nanocomputing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
- Bennett, C. H. (1973). Logical reversibility of computation. *IBM journal of Research and Development*, 17(6):525--532.
- Bennett, C. H. (1989). Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766--776.
- Bérut, A., Arakelyan, A., Petrosyan, A., Ciliberto, S., Dillenschneider, R., and Lutz, E. (2012). Experimental verification of landauer's principle linking information and thermodynamics. *Nature*, 483(7388):187--189.

- Blair, E. P. and Lent, C. S. (2013). Environmental decoherence stabilizes quantum-dot cellular automata. *Journal of Applied Physics*, 113(12):124302.
- Blair, E. P., Tóth, G., and Lent, C. S. (2018). Entanglement loss in molecular quantum-dot qubits due to interaction with the environment. *Journal of Physics: Condensed Matter*, 30(19):195602.
- Chabi, A. M., Roohi, A., Khademolhosseini, H., Sheikhfaal, S., Angizi, S., Navi, K., and DeMara, R. F. (2017). Towards ultra-efficient qca reversible circuits. *Microprocessors and Microsystems*, 49:127--138.
- Chaves, J. F., Ribeiro, M. A., Silva, L. M., de Assis, L. M. B. C., Torres, M. S., and Vilela Neto, O. P. (2018a). Energy efficient QCA circuits design: simulating and analyzing partially reversible pipelines. *Journal of Computational Electronics*, 17(1):479--489. ISSN 1572-8137.
- Chaves, J. F., Ribeiro, M. A., Torres, F. S., and Neto, O. P. V. (2019). Designing partially reversible field-coupled nanocomputing circuits. *IEEE Transactions on Nanotechnology*, 18:589--597.
- Chaves, J. F., Ribeiro, M. A., Torres, F. S., and Vilela Neto, O. P. (2018b). Enhancing fundamental energy limits of field-coupled nanocomputing circuits. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE.
- Cirillo, G. A., Turvani, G., and Graziano, M. (2019). A quantum computation model for molecular nanomagnets. *IEEE Transactions on Nanotechnology*, 18:1027--1039.
- Conte, T., DeBenedictis, E., Ganesh, N., Hylton, T., Still, S., Strachan, J. P., Williams, R. S., Alemi, A., Altenberg, L., Crooks, G., et al. (2019). Thermodynamic computing. *arXiv preprint arXiv:1911.01968*.
- Cross, T. (2016). After moore's law—double, double, toil and trouble. *The Economist, Technology Quarterly, Quarter*, 1.
- Das, J. C. and De, D. (2017). Nanocommunication network design using qca reversible crossbar switch. *Nano Communication Networks*, 13:20--33.
- DeBenedictis, E. P. (2016). The boolean logic tax. *Computer*, 49(4):79--82. ISSN 0018-9162.
- DeBenedictis, E. P. (2017). It's time to redefine moore's law again. *Computer*, 50(2):72-75.

- DeBenedictis, E. P., Frank, M. P., Ganesh, N., and Anderson, N. G. (2016). A path toward ultra-low-energy computing. In *Rebooting Computing (ICRC), IEEE International Conference on*, pages 1--8. IEEE.
- Dennard, R. H., Gaensslen, F. H., Rideout, V. L., Bassous, E., and LeBlanc, A. R. (1974). Design of ion-implanted mosfet's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256--268.
- Deutsch, D. (1989). Quantum computational networks. *Proc. R. Soc. Lond. A*, 425(1868):73--90.
- Eichwald, I., Breitzkreutz, S., Ziemys, G., Csaba, G., Porod, W., and Becherer, M. (2014). Majority logic gate for 3d magnetic computing. *Nanotechnology*, 25(33):335202.
- Ercan, I. and Anderson, N. G. (2011). Heat dissipation bounds for nanocomputing: Theory and application to qca. In *2011 11th IEEE International Conference on Nanotechnology*, pages 1289--1294. IEEE.
- Ercan, I. and Anderson, N. G. (2013). Heat dissipation in nanocomputing: lower bounds from physical information theory. *Nanotechnology, IEEE Transactions on*, 12(6):1047--1060.
- Formigoni, R. E., Araujo Vieira, L. L., Vilela Neto, O. P., Ferreira, R., and Nacif, J. A. M. (2020). Evaluating nanomagnetic logic circuit layouts using different clock schemes. *Analog Integrated Circuits and Signal Processing*.
- Formigoni, R. E., Ferreira, R. S., and Nacif, J. A. M. (2019). Ropper: a placement and routing framework for field-coupled nanotechnologies. In *2019 32nd Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 1--6. IEEE.
- Frank, M. P. (2005). Approaching the physical limits of computing. In *Multiple-Valued Logic, 2005. Proceedings. 35th International Symposium on*, pages 168--185. IEEE.
- Frank, M. P. (2017). Foundations of generalized reversible computing. In *International Conference on Reversible Computation*, pages 19--34. Springer.
- Frank, M. P. (2018a). Back to the future: The case for reversible computing. *arXiv preprint arXiv:1803.02789*.
- Frank, M. P. (2018b). Physical foundations of landauer's principle. In *International Conference on Reversible Computation*, pages 3--33. Springer.

- Fredkin, E. and Toffoli, T. (1982). Conservative logic. *International Journal of Theoretical Physics*, 21(3):219--253.
- Frost-Murphy, S. E. (2009). *Reversibility for Nanoscale Systems*. PhD dissertation, University of Notre Dame.
- Haider, M. B., Pitters, J. L., DiLabio, G. A., Livadaru, L., Mutus, J. Y., and Wolkow, R. A. (2009). Controlled coupling and occupation of silicon atomic quantum dots at room temperature. *Physical review letters*, 102(4):46805.
- Hong, J., Lambson, B., Dhuey, S., and Bokor, J. (2016). Experimental test of landauer's principle in single-bit operations on nanomagnetic memory bits. *Science Advances*, 2(3):e1501492.
- Hong, J., Stone, M., Navarrete, B., Luongo, K., Zheng, Q., Yuan, Z., Xia, K., Xu, N., Bokor, J., You, L., et al. (2018a). 3d multilevel spin transfer torque devices. *Applied Physics Letters*, 112(11):112402.
- Hong, J., Stone, M., Navarrete, B., Luongo, K., Zheng, Q., Yuan, Z., Xia, K., Xu, N., Bokor, J., You, L., and Khizroev, S. (2018b). 3D multilevel spin transfer torque devices. *Applied Physics Letters*, 112(11):112402.
- Huff, T., Labidi, H., Rashidi, M., Livadaru, L., Dienel, T., Achal, R., Vine, W., Pitters, J., and Wolkow, R. A. (2018). Binary atomic silicon logic. *Nature Electronics*, 1(12):636--643.
- Imre, A., Csaba, G., Ji, L., Orlov, A., Bernstein, G., and Porod, W. (2006). Majority logic gate for magnetic quantum-dot cellular automata. *Science*, 311(5758):205--208.
- Kianpour, M. and Sabbaghi-Nadooshan, R. (2017). Novel 8-bit reversible full adder/subtractor using a qca reversible gate. *Journal of Computational Electronics*, 16(2):459--472.
- Kolmer, M., Zuzak, R., Dridi, G., Godlewski, S., Joachim, C., and Szymonski, M. (2015). Realization of a quantum hamiltonian boolean logic gate on the Si (001): H surface. *Nanoscale*, 7(29):12325--12330.
- Lambson, B., Carlton, D., and Bokor, J. (2011). Exploring the thermodynamic limits of computation in integrated systems: magnetic memory, nanomagnetic logic, and the landauer limit. *Physical review letters*, 107(1):010604.

- Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM journal of research and development*, 5(3):183--191.
- Landauer, R. (1989). Nanostructure physics: fashion or depth. *Nanostructure Physics and Fabrication*, pages 17--30.
- Lent, C. S. (2019). Information and entropy in physical systems. In *Energy Limits in Computation*, pages 1--63. Springer.
- Lent, C. S., Henderson, K. W., Kandel, S. A., Corcelli, S. A., Snider, G. L., Orlov, A. O., Kogge, P. M., Niemier, M. T., Brown, R. C., Christie, J. A., et al. (2016). Molecular cellular networks: a non von neumann architecture for molecular electronics. In *Rebooting Computing (ICRC), IEEE International Conference on*, pages 1--7. IEEE.
- Lent, C. S., Liu, M., and Lu, Y. (2006). Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling. *Nanotechnology*, 17(16):4240.
- Lent, C. S., Orlov, A. O., Porod, W., and Snider, G. L. (2018). *Energy Limits in Computation: A Review of Landauer's Principle, Theory and Experiments*. Springer.
- Lent, C. S. and Snider, G. L. (2014). The development of quantum-dot cellular automata. In *Field-Coupled Nanocomputing*, pages 3--20. Springer.
- Lent, C. S. and Tougaw, P. D. (1997a). A device architecture for computing with quantum dots. *Proceedings of the IEEE*, 85(4):541--557.
- Lent, C. S. and Tougaw, P. D. (1997b). A device architecture for computing with quantum dots. *Proceedings of the IEEE*, 85(4):541--557.
- Lent, C. S., Tougaw, P. D., Porod, W., and Bernstein, G. H. (1993). Quantum cellular automata. *Nanotechnology*, 4(1):49.
- Li, M. and Vitanyi, P. (1996). Reversibility and adiabatic computation: trading time and space for energy. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 452, pages 769--789. The Royal Society.
- Ma, X., Huang, J., Metra, C., and Lombardi, F. (2008). Reversible gates and testability of one dimensional arrays of molecular qca. *Journal of Electronic Testing*, 24(1-3):297--311.



- Manipatrumi, S., Nikonov, D. E., Lin, C.-C., Gosavi, T. A., Liu, H., Prasad, B., Huang, Y.-L., Bonturim, E., Ramesh, R., and Young, I. A. (2019). Scalable energy-efficient magnetoelectric spin-orbit logic. *Nature*, 565(7737):35--42.
- Martini, L., Pancaldi, M., Madami, M., Vavassori, P., Gubbiotti, G., Tacchi, S., Hartmann, F., Emmerling, M., Höfling, S., Worschech, L., et al. (2016). Experimental and theoretical analysis of landauer erasure in nano-magnetic switches of different sizes. *Nano Energy*, 19:108--116.
- Mollick, E. (2006). Establishing moore's law. *IEEE Annals of the History of Computing*, 28(3):62--75.
- Moore, G. E. (1965). Cramming more components onto integrated circuits.
- Morrison, M. and Ranganathan, N. (2014). Synthesis of dual-rail adiabatic logic for low power security applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(7):975--988.
- Neri, I. and López-Suárez, M. (2016). Heat production and error probability relation in landauer reset at effective temperature. *Scientific reports*, 6:34039.
- Neutzling, A., Marranghello, F. S., Matos, J. M., Reis, A., and Ribas, R. P. (2019). Maj-n logic synthesis for emerging technology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
- Neutzling, A., Martins, M. G., Callegaro, V., Reis, A. I., and Ribas, R. P. (2017). A simple and effective heuristic method for threshold logic identification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(5):1023--1036.
- Neutzling, A., Matos, J. M., Mishchenko, A., Reis, A., and Ribas, R. P. (2018). Effective logic synthesis for threshold logic circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(5):926--937.
- Ng, S. S. H., Retallick, J., Chiu, H. N., Lupoiu, R., Livadaru, L., Huff, T., Rashidi, M., Vine, W., Dienel, T., Wolkow, R. A., et al. (2020). Siqad: A design and simulation tool for atomic silicon quantum dot circuits. *IEEE Transactions on Nanotechnology*, 19:137--146.
- Orlov, A. O., Lent, C. S., Thorpe, C. C., Boechler, G. P., and Snider, G. L. (2012). Experimental test of landauer's principle at the sub-kbt level. *Japanese Journal of Applied Physics*, 51(6):06FE10.

- Ottavi, M., Pontarelli, S., DeBenedictis, E. P., Salsano, A., Frost-Murphy, S., Kogge, P. M., and Lombardi, F. (2011). Partially reversible pipelined qca circuits: combining low power with high throughput. *Nanotechnology, IEEE Transactions on*, 10(6):1383–1393.
- Parrondo, J. M., Horowitz, J. M., and Sagawa, T. (2015). Thermodynamics of information. *Nature physics*, 11(2):131–139.
- Pitters, J. L., Livadaru, L., Haider, M. B., and Wolkow, R. A. (2011). Tunnel coupled dangling bond structures on hydrogen terminated silicon surfaces. *The Journal of chemical physics*, 134(6):064712.
- Preskill, J. (2018). Quantum computing in the nisq era and beyond. *arXiv preprint arXiv:1801.00862*.
- Ramsey, J. S. and Blair, E. P. (2017). Operator-sum models of quantum decoherence in molecular quantum-dot cellular automata. *Journal of Applied Physics*, 122(8):084304.
- Ribeiro, M. A., Carvalho, I. A., Chaves, J. F., and Neto, O. P. V. (2018a). Improving Energy Efficiency on Partially Reversible Pipelined QCA Circuits. In *2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI)*.
- Ribeiro, M. A., Carvalho, I. A., Chaves, J. F., and Neto, O. P. V. (2020). Optimal energy efficiency and throughput on partially reversible pipelined QCA circuits. *IEEE Design & Test*.
- Ribeiro, M. A., Carvalho, I. A., Chaves, J. F., Pappa, G. L., and Neto, O. P. V. (2018b). Improving Energy Efficiency of Field-Coupled Nanocomputing Circuits by Evolutionary Synthesis. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.
- Riener, H., Testa, E., Haaswijk, W., Mishchenko, A., Amarú, L., De Micheli, G., and Soeken, M. (2019). Logic optimization of majority-inverter graphs. In *MBMV 2019; 22nd Workshop-Methods and Description Languages for Modelling and Verification of Circuits and Systems*, pages 1–4. VDE.
- Roohi, A., Zand, R., Angizi, S., and Demara, R. F. (2016). A parity-preserving reversible qca gate with self-checking cascadable resiliency. *IEEE Transactions on emerging topics in computing*.

- Saeed, S. M., Zulehner, A., Wille, R., Drechsler, R., and Karri, R. (2019). Reversible circuits: Ic/ip piracy attacks and countermeasures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(11):2523--2535.
- Saeedi, M. and Markov, I. L. (2013). Synthesis and optimization of reversible circuits—a survey. *ACM Computing Surveys (CSUR)*, 45(2):21.
- Sen, B., Dutta, M., Some, S., and Sikdar, B. K. (2014). Realizing reversible computing in qca framework resulting in efficient design of testable alu. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 11(3):30.
- Silva, P. A. R., Neto, O. P. V., and Nacif, J. A. M. (2019). Toward nanometric scale integration: an automatic routing approach for nml circuits. In *2019 32nd Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 1--6. IEEE.
- Singh, G., Sarin, R., and Raj, B. (2017). Design and analysis of area efficient qca based reversible logic gates. *Microprocessors and Microsystems*, 52:59--68.
- Soeken, M., Roetteler, M., Wiebe, N., and De Micheli, G. (2018). Lut-based hierarchical reversible logic synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(9):1675--1688.
- Srivastava, S., Asthana, A., Bhanja, S., and Sarkar, S. (2011). Qcapro—an error-power estimation tool for qca circuit design. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 2377--2380. IEEE.
- Stearns, K. J. and Anderson, N. G. (2013). Throughput-dissipation tradeoff in partially reversible nanocomputing: A case study. In *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 101--105. ISSN 2327-8218.
- Testa, E., Soeken, M., Amar, L. G., and De Micheli, G. (2018a). Logic synthesis for established and emerging computing. *Proceedings of the IEEE*, 107(1):165--184.
- Testa, E., Soeken, M., Amaru, L. G., Haaswijk, W., and De Micheli, G. (2018b). Mapping monotone boolean functions into majority. *IEEE Transactions on Computers*, 68(5):791--797.
- Testa, E., Zografos, O., Soeken, M., Vaysset, A., Manfrini, M., Lauwereins, R., and De Micheli, G. (2017). Inverter propagation and fan-out constraints for beyond-CMOS majority-based technologies. In *VLSI (ISVLSI), 2017 IEEE Computer Society Annual Symposium on*, pages 164--169. IEEE.

- Thapliyal, H. and Ranganathan, N. (2010). Reversible logic-based concurrently testable latches for molecular qca. *Nanotechnology, IEEE Transactions on*, 9(1):62--69.
- Torres, F. S., Niemann, P., Wille, R., and Drechsler, R. (2019). Near zero-energy computation using quantum-dot cellular automata. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 16(1):1--16.
- Torres, F. S., Silva, P. A., Fontes, G., Walter, M., Nacif, J. A. M., Ferreira, R. S., Neto, O. P. V., Chaves, J. F., Wille, R., Niemann, P., et al. (2020). On the impact of the synchronization constraint and interconnections in quantum-dot cellular automata. *Microprocessors and Microsystems*, page 103109.
- Torres, F. S., Wille, R., Niemann, P., and Drechsler, R. (2018). An energy-aware model for the logic synthesis of quantum-dot cellular automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
- Tougaw, P. D. and Lent, C. S. (1994). Logical devices implemented using quantum cellular automata. *Journal of Applied physics*, 75(3):1818--1825.
- Versluis, R. and Hagen, C. (2020). Quantum computers scale up: Constructing a universal quantum computer with a large number of qubits will be hard but not impossible. *IEEE Spectrum*, 57(4):24--29.
- Vieri, C., Ammer, M. J., Frank, M., Margolus, N., and Knight, T. (1998). A fully reversible asymptotically zero energy microprocessor. In *Power Driven Microarchitecture Workshop*, pages 138--142. Citeseer.
- Walter, M., Wille, R., Große, D., Torres, F. S., and Drechsler, R. (2018). An exact method for design exploration of quantum-dot cellular automata. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 503--508. IEEE.
- Walter, M., Wille, R., Große, D., Torres, F. S., and Drechsler, R. (2019). Placement and routing for tile-based field-coupled nanocomputing circuits is np-complete (research note). *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(3):1--10.
- Walus, K., Dimitrov, V., Jullien, G. A., and Miller, W. C. (2003). Qcadesigner: A cad tool for an emerging nano-technology. In *Micronet Annual Workshop*, pages 292--294.

- Wille, R., Haghparast, M., Adarsh, S., and Tanmay, M. (2019). Towards hdl-based synthesis of reversible circuits with no additional lines. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1--7. IEEE.
- Yang, S. (1991). Logic synthesis and optimization benchmarks user guide version 3.0.
- Ye, P., Ernst, T., and Khare, M. V. (2019). The last silicon transistor: Nanosheet devices could be the final evolutionary step for Moore's Law. *IEEE Spectrum*, 56(8):30-35.
- Younis, S. G. and Knight, Jr., T. F. (1993). Practical implementation of charge recovering asymptotically zero power cmos. In *Proceedings of the 1993 Symposium on Research on Integrated Systems*, pages 234--250, Cambridge, MA, USA. MIT Press.
- Zhou, S. and Blair, E. P. (2020). Non-markovian models of environmentally-driven disentanglement in molecular charge qubits. *Journal of Applied Physics*, 127(8):084303.
- Zografos, O., Manfrini, M., Vaysset, A., Sorée, B., Ciubotaru, F., Adelman, C., Lauwereins, R., Raghavan, P., and Radu, I. P. (2017). Exchange-driven magnetic logic. *Scientific reports*, 7(1):12154.
- Zulehner, A., Frank, M. P., and Wille, R. (2019). Design automation for adiabatic circuits. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 669--674.