

Unconventional Computing Review

Ioannis Savvaidis

Abstract—As we enter the Fourth Industrial Revolution, characterized by the fusion of physical, digital, and biological systems, traditional computing paradigms are increasingly constrained by the limitations of Moore’s Law and Dennard Scaling. This review delves into the historical context and evolution of computing, highlighting the urgent need for innovative paradigms beyond conventional methods. Specifically, it explores the potential of unconventional computing approaches such as reversible computing, optical computing, and soliton-based computing. By harnessing principles from these diverse fields, these paradigms promise to overcome current technological limitations and revolutionize industries, including artificial intelligence, optimization, communication, and cryptography. This survey underscores how exploiting the intrinsic physical properties of materials can lead to the development of more powerful and efficient computing systems, paving the way for the next generation of technological advancements.

Index Terms—Unconventional computing, Reversible computing, Optical computing, Soliton computing

I. INTRODUCTION

A. Historical Background

From the beginning of our known time, human history is constantly changing. While this change can have good or bad effects on the cosmos, we can observe a definite technological evolution in how we understand nature during the evolution of time. In the film “2001: A Space Odyssey”, Stanley Kubrick explores and more precisely understands a sequence of revolutionary periods in knowledge and advancements with a pattern of a mysterious “alien monolith” appearance, followed by intelligence advancements of the species that observed it.

We can observe the remarkable precision of this film with real life, based on our history, but more importantly, for its prediction of the nearest future. Scientists have categorized human history’s most major evolution and technological advancements into four Industrial Revolutions, as depicted in Figure 1.

Navigating the next industrial revolution		
Revolution	Year	Information
1	1784	Steam, water, mechanical production equipment
2	1870	Division of labour, electricity, mass production
3	1969	Electronics, IT, automated production
4	?	Cyber-physical systems

WORLD
ECONOMIC
FORUM
COMMITTED TO IMPROVING THE WORLD

Fig. 1. Four Industrial Revolutions

We are now witnessing the rise of a Fourth Industrial Revolution, which builds upon the Third Industrial Revolution, also known as the digital revolution, that began in the mid-20th century. This new era is marked by a convergence of technologies that blur the physical, digital, and biological boundaries. There are three key reasons why today’s transformations signify the emergence of a distinct Fourth Industrial Revolution rather than a mere extension of the Third: velocity, scope, and systemic impact. The current rate of technological breakthroughs is unprecedented in history, advancing at an exponential rather than a linear pace. Furthermore, it is causing widespread disruption across nearly every industry and country. The extensive and profound nature of these changes is transforming entire systems of production, management, and governance [1].

But let us focus on the Third one to understand what enabled this exponential growth. While the reasons can be many, the exponential growth of technology in terms of computing power dictated the significant advances in sciences, Arts, and Culture. This exponential growth has predicted and sustained a tremendous tech industry, pushing the limits every year according to the well-known Moore’s Law. Currently, society depends on the rapid and affordable technology that Moore predicted in 1965.

The age of digital electronics began in 1947 when a research team at Bell Labs designed the first transistor, and the first commercially available Integrated Circuit (IC or Chip) appeared in 1961. In 1964, some chips had as many as 32 transistors, and when Moore wrote his article in 1965, a chip in his lab had 64. Hence, he noticed the complexity of the IC increment rate. A new economy was introduced based on this extraordinary rate of innovation. Apart from the economic impact, the computerization that started around the 90s reshaped the world. Productivity growth, linked directly with computer usage, established computers as the most important single technology, while this productivity growth led to improved living standards. Thus, we have built a tremendous tech industry to scale, spread, and embed technology in our day-to-day lives [2].

Moore’s Law is a techno-economic model that has enabled the industry to double the performance and functionality of digital electronics approximately every two years, within fixed cost and power (Figure 2). More precisely, it postulates that the level of chip complexity that can be manufactured for minimal cost is an exponential function that doubles in a period. It predicts that the cost of making any given integrated circuit at optimal transistor density levels is constant in time [3].

The optimal component density, for any given period t :

$$C_t = 2C_{t-1}, \quad (1)$$

The Manufacturing cost per component M in period t:

$$M_t = \frac{M_{t-1}}{2} \quad (2)$$

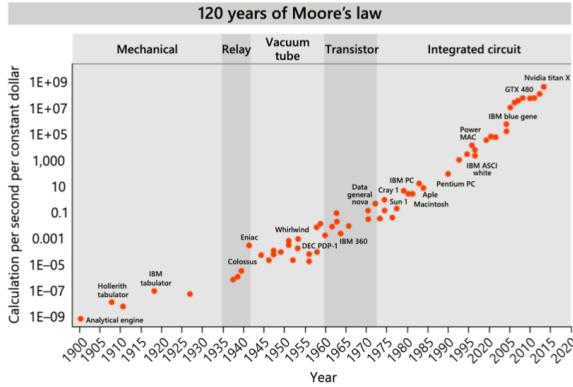


Fig. 2. 120 years of Moore's Law

Moore's Law relies on the CMOS technology, and the prediction is based on the transistor's physical length scaling. Advances in Semiconductor Lithography enabled this scaling as the miniaturization of electronics, resulting in exponential computing power and affordable commercial tech. Currently, in 2024, we are counting transistors in hundreds of billions in typical ICs and measuring the size of transistors in one-digit nanometers. As transistors reach their atomic size limit, physical and economic constraints make it difficult to sustain Moore's Law. Challenges like quantum effects, heat dissipation, material limitations, and increasing fabrication costs required fundamental advances in computing, marking a new era of computing [4].

Since the 1990s, the semiconductor industry has published the International Technology Roadmap for Semiconductors (ITRS) every two years to coordinate industry-wide efforts for advancement. However, in 2017, updates to the ITRS were discontinued, signaling the approaching end of conventional CMOS transistor scaling (Figure 3).

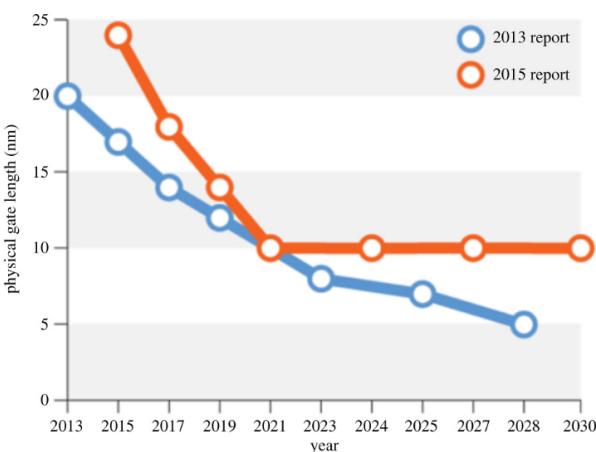


Fig. 3. ITRS Roadmap

Apart from transistor scaling, another scaling area reached a limit. Dennard Scaling, named after Robert H. Dennard, first described the concept in a 1974 paper. Dennard Scaling refers to a critical principle in semiconductor design and microelectronics. It states that as transistors get smaller, their power density remains constant, meaning that the power consumption does not increase despite the transistors being more numerous and closer together. This is achieved by reducing the voltage and current in proportion to the linear dimensions of the transistors. Dennard Scaling allows for higher performance and efficiency in smaller spaces. However, in the mid-2000s, Dennard scaling failed, resulting in a stagnation of clock rate increases. The maximum clock speed stabilized at approximately 3–4 GHz, with power consumption peaking in the range of a few hundred watts (Figure 4). This limitation on clock frequency questions if there are innovative alternatives to the fixed-frequency clocking design strategy that has dominated IC design for decades. Moreover, it is challenging to determine if dynamic-frequency clocking is beneficial in applications whenever and wherever possible.

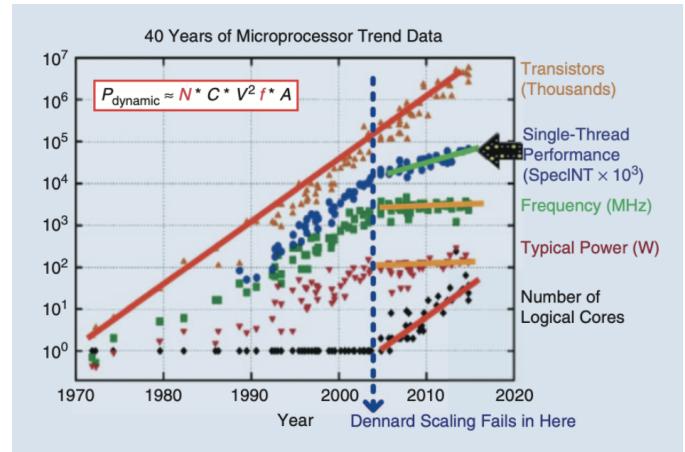


Fig. 4. Clock Speed Stabilization

Moore also predicted this limitation in 2001: "I've learned to live with the term. But it's really not a law; it's a prediction. No exponential runs forever. The key has always been our ability to shrink dimensions and we will soon reach atomic dimensions, which are an absolute limit." [5]. The industry continues to enhance device functionalities, summarized in the following strategies (Figure 5) [6].

Space

- **More Moore**

A near-term strategy focuses on continuous transistor scaling and miniaturization. Initially, as transistors shrank, they also became faster and more energy-efficient. However, Dennard Scaling broke down, halting clock speed increases due to heat dissipation issues. To continue performance improvements, the industry shifted to multi-core architectures and power-efficient designs, alongside innovations like strained silicon and 3D transistors, to use a large number of basic processing elements more efficiently as a whole.

• More Than Moore

A mid-term strategy integrating various technologies (e.g., sensors, memory, power management, communication modules) into a single system to enhance overall capabilities beyond individual transistors, as seen in modern smart home hubs (IoT devices). This approach also includes heterogeneous integration, which combines different technologies on a single chip using advanced packaging techniques for improved functionality and performance, exemplified by modern smartphones with integrated SoCs, sensors, modems, image processing modules, and PMICs. Additionally, More Than Moore tailors accelerator chip designs to specific applications, optimizing power consumption and processing speed, as seen in Application Specific Integrated Circuits (ASICs) such as GPUs, TPUs, and NPUs. This approach highlights that technological advancements are not solely driven by increased computing power but by enhanced system capabilities and tailored solutions.

• Beyond Moore

A long-term strategy exploring CMOS alternatives by moving beyond CMOS with advancements in ICs, with stack chips and Chiplet technology, to pack more transistors efficiently, like 3D chip stacking. In addition, it is investigating materials like graphene and carbon nanotubes, which are being considered for transistor channels and interconnects due to their superior properties, to surpass silicon limitations. It also includes novel computational paradigms as alternatives to traditional silicon-based transistors, such as quantum computing, neuromorphic computing, and optical computing. These approaches aim to overcome the limitations of conventional scaling by leveraging new technologies and architectures.

Time

The above strategies focus on space in terms of computation power growth. There are also two additional considerations for these paths to investigate more computing power from the time perspective. **Clock Moore** investigates dynamic-frequency clocking to enhance computing power. Meanwhile, **Rate Moore** proposes using time rather than space to encode information, suggesting new computational models prioritizing timing for increased efficiency. Over the past five decades, the semiconductor industry has primarily relied on electric charges (electron movement, represented as voltage and/or current) to encode information. This method requires matter and, consequently, space, as the amount of charge is virtually proportional to the size of the space [6].

B. Computing in a Nutshell

Standard computing, or in other words, conventional computing, is based on highly developed theories of classical computing and the extraordinary engineering of CMOS technology. Unconventional Computing is going beyond that path, extending the barriers between philosophy and engineering by exploring different physical systems and information processing paradigms. Ultimately, both computing models should

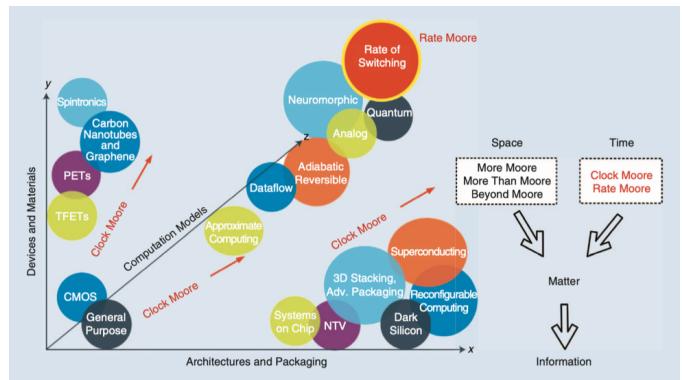


Fig. 5. Future of Computing

question the confidence of using a system as a computer. To address this question, we need to have a formalized computing model to decide whether the system is a computer. When we move beyond Classical Computing, the foundations of Information Science continue to be addressed. Is the human mind a computer? These foundations are the choreography between Information, Computation, and Computer [7].

Information is always tied to a physical representation. Maxwell's Demon [8] illustrates the profound connection between information and thermodynamics. While the demon's ability to decrease entropy appears to violate the second law of thermodynamics, the resolution lies in realizing that information processing has thermodynamic costs. In the 1960s, Rolf Landauer formalized that erasing information in a computing system must produce a corresponding increase in entropy, typically manifesting as heat dissipation [9]. This principle helped to resolve the paradox by showing that the demon's measurements and memory erasures result in an overall increase in entropy, thus preserving the second law of thermodynamics. These advances and experiments led to accepting information as a physical entity. Extending the formalization further, "the laws of physics are essentially algorithms for calculations," as cited by Landauer [10].

Since we can describe our understanding of physical objects with a theory and mathematical abstract objects, we can calculate and predict their evolution by performing computations. As Georgios Ch. Sirakoulis mentions, "Computing is understanding." [11]. We use computers in our everyday lives, and we let them perform these computing operations for us because we first understand them, model them, and then fabricate them. Abstraction/Representation (AR) theory (Horsman et al. 2014) has been developed to formalize the definition and requirements of a physical system that can be used as a computer [12].

A computer is a physical system with constituent parts and internal interactions that transition from one physical state to another. It must be capable of encoding and decoding information and performing at least one fundamental dynamical operation. Central to this definition is the representation relation, which maps physical systems to mathematical objects, allowing comparisons and defining when a physical process is utilized for computation. The theory distinguishes the abstract

and physical spaces and connects them solely by the relation of the directed representation.

Furthermore, AR Theory introduces the notion of computational entities (physical entities that locate the representation relation), which is essential for establishing when computing occurs within physical systems. These entities are responsible for the encoding and decoding steps, crucial for translating abstract data into physical states and vice versa (Humans or any Intelligent system). A computer is used to predict the outcome of an abstract evolution. This predictive capability distinguishes computing from other physical process evolution, as the system must reliably perform computations that produce predictable results.

Stepney and Kendon extended the AR model to include a Representational Entity (RE), which supports the representation relation between physical and abstract entities [13].

The full compute cycle (Figure 6) involves representing the representational entity's desired physical state P_{RE} as an abstract model $m_{P_{RE}}$, encoding this to a computational model m_{P_c} , and instantiating it into the physical computer state P_c . The physical computer then evolves to the state P'_c , with the physical evolution H_c . The new state is represented as an abstract computational solution m'_{P_c} . Finally, this is decoded to the abstract problem solution $m'_{P_{RE}}$, which is the result of abstract evolution C_c , modeling the final state of the RE. Each step should approximately commute, ensuring consistency between the representational entity and the physical computer throughout the compute cycle.

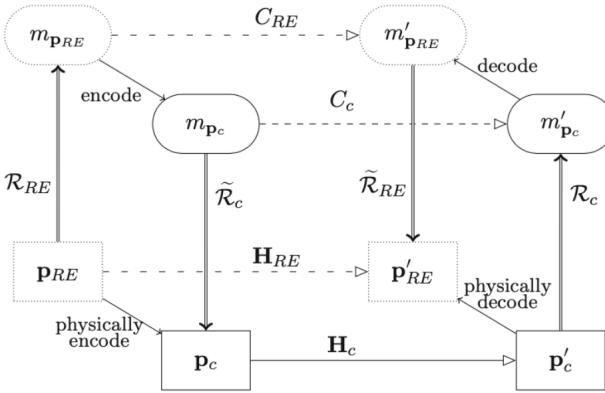


Fig. 6. Full Compute Cycle of AR Theory with RE extension

II. UNCONVENTIONAL COMPUTING

As the demand for more powerful and efficient computing systems grows, the cessation of Moore's Law, the constraints imposed by Dennard Scaling, and the Turing barrier have become increasingly apparent. Researchers are exploring Beyond Moore strategies to address these challenges, including Unconventional Computing, which presents promising alternatives to traditional digital computing paradigms.

Unconventional Computing encompasses diverse computational approaches that depart from reliance on binary logic, silicon-based technology, and von Neumann architectures.

Instead, it leverages principles from various fields, such as quantum mechanics, neuromorphic engineering, optics, and spintronics. These innovative approaches can potentially revolutionize industries ranging from artificial intelligence and optimization to communication and cryptography.

The essence of Unconventional Computing lies in exploiting the physics of materials directly for computational operations. This approach involves representing information as physical quantities and utilizing materials' inherent physical dynamics and evolution for computation.

For instance, an origami computer (Figure 7), as depicted in a Quanta Magazine article (Quanta Magazine, 2024), illustrates how the properties of physical materials can be harnessed to perform computational tasks, embodying the core idea of Unconventional Computing.

The field seeks to develop new computing architectures using chemical, physical, and biological mechanisms and aims to uncover entirely different ways to compute, broadening our understanding of computation. A few paradigms are Quantum, Optical, Soliton, Spintronics, Reservoir, and DNA Computing. In recent years, heavy academic and experimental research has led to new computational models with cross-disciplines, combining multiple models of different scientific domains [14].

The origins of Unconventional Computing trace back to the mid-20th century. Notable pioneers include Alan Turing, whose work on "The Chemical Basis of Morphogenesis" explored how natural patterns could inspire computational models. The 1980s marked significant advancements with Richard Feynman's foundational contributions to quantum computing, particularly his 1982 paper, "Simulating Physics with Computers," which laid the groundwork for utilizing quantum mechanics in computation. The 1990s saw further developments with Carver Mead's pioneering work in neuromorphic engineering, as detailed in his seminal paper, "Neuromorphic Electronic Systems" (1990). These early explorations laid the groundwork for today's diverse and interdisciplinary approaches, driving the continuous evolution of Unconventional Computing [7].

In the following sections, we will review various unconventional computing models, categorized by their computing paradigms, and explore how they can pave the way for the next generation of computing technologies. Due to this master's program, we will cover some of them, leaving Quantum Computing out.

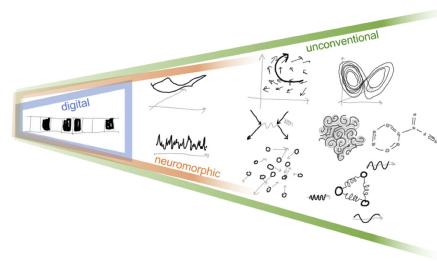


Fig. 7. Road to Unconventional Computing

III. REVERSIBLE COMPUTING

A. Introduction

Reversible computing is a paradigm where computations can be reversed, meaning the system's state can be uniquely traced backward from output to input. This backward determinism ensures that every computational state has precisely one preceding state, allowing the process to be time-reversed step-by-step. This concept is crucial for reducing energy dissipation in computing systems, as conventional irreversible computations inherently lose information, generating heat. Reversible computing is thus fundamental to creating more energy-efficient systems, particularly at the nanoscale, where traditional heat management becomes increasingly challenging.

The importance of reversible computing is underscored by Landauer's principle, which states that the minimum energy E needed to erase one bit of information is proportional to the temperature T at which the system is operating and to Boltzmann constant K_B :

$$E \geq K_B T \ln 2 \quad (3)$$

This principle highlights the inherent energy cost of irreversible operations. By contrast, reversible computing allows for computations where no information is lost, theoretically enabling computations to occur without energy dissipation beyond what is required to maintain physical coherence. Bennett further developed this idea, demonstrating that any irreversible computation can be simulated by a reversible one without leaving any garbage information, thus eliminating energy dissipation due to information erasure [15].

Reversible computing is not only a theoretical construct but has practical implications and applications in emerging technologies such as quantum computing. These technologies benefit from the reduced heat generation and increased efficiency offered by reversible computing principles. The concept originated from studies on the thermodynamics of computation and has since evolved into a significant area of research aimed at developing energy-efficient and sustainable computing technologies.

B. Abstract Realization

Reversible computing is based on the theoretical foundation of bijective functions, ensuring each output has a unique input (one-to-one). In essence, it is possible to recover the complete initial state of the system, having only the final state, which to some extent establishes a time-reversible computational process, and no information is lost. These functions are represented as Logic Gates and must follow the same number of outputs as inputs, and the mapping of input to output states must be one-to-one.

Reversible Logic Gates

Reversibility in computing means no information about computational states is lost, allowing the recovery of earlier stages by computing backward. This is known as logical reversibility. Reversible computing impacts digital logic design, requiring

reversible logic elements to recover input states from outputs. The one-bit NOT gate (Figure 8) is reversible in classical (conventional) Logic Gates. Based on this is the building block of more advanced reversible gates. Extending to a two-bit CNOT gate (Controlled-NOT), one bit is the controller, and the other bit is the target. If the control bit is one, a NOT gate is performed on the target bit, and if the control bit is zero, the target bit is left unchanged. By applying the CNOT twice on the same bits, we will reverse the system to the original state (4). In contradiction, the AND Gate is not reversible since we cannot revert the system's final state of one bit to determine the two-bit inputs of the AND Gate.

$$(a, b) \rightarrow (a, a \oplus b) \rightarrow (a, a \oplus b \oplus b) = (a, b) \quad (4)$$

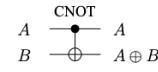


Fig. 8. CNOT Gate

Using CNOTs in a specific topology, we can build a SWAP Gate (Figure 9), which is very important to overcome the wiring swapping of the physical circuit.

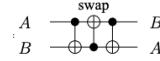


Fig. 9. SWAP Gate

Fredkin and Toffoli invented the two most commonly used three-bit reversible gates.

• Toffoli Gate

Toffoli Gate or controlled-controlled-NOT Gate (Figure 10) has tree bits input. The first two bits are the controllers, and the third is the target bit. If both control bits (A and B) are one, then a NOT gate is performed on the target bit (C); otherwise, no action is performed.

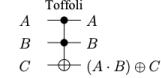


Fig. 10. Toffoli Gate

• Fredkin Gate

Fredkin Gate, or controlled-SWAP gate (Figure 10), has three bits of input. The first is the controller, and the other two are the targets. If the control bit (A) is one, the two targets (B and C) have their values swapped, and if the control is zero, the targets are unaffected.

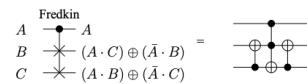


Fig. 11. Fredkin Gate

Reversible Circuits

Any computable circuit or equation can be reduced to a set of Toffoli or Fredkin gates. Smaller gates, such as the CNOT and NOT, can be simulated by setting one or two of the inputs to the gate to zero or one, as appropriate. These gates are universal, meaning any logical function can be implemented only if additional constant inputs and garbage outputs are allowed. In reversible circuits, direct fan-out is not permitted because one-to-many mapping isn't reversible. Fan-out is achieved using additional gates. Designing reversible circuits requires minimizing the number of reversible logic gates. Several parameters determine the complexity and performance of these circuits [16]:

- **Number of Reversible Gates (N):** Total gates used in the circuit.
- **Number of Constant Inputs (CI):** Inputs maintained constant at 0 or 1 for synthesizing the logical function.
- **Number of Garbage Outputs (GO):** Unused outputs necessary for achieving reversibility.
- **Quantum Cost (QC):** Cost of primitive gates needed to realize the circuit.

For example, emulating classical AND and OR with Toffoli Gates (Figure 12):

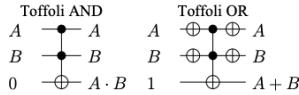


Fig. 12. Toffoli AND and OR Gates

There are many other Reversible Gates, such as Feynman and Peres Gate. A reversible circuit (Figure 14) is a logic circuit in which all gates are reversible and interconnected without explicit fanouts or loops.

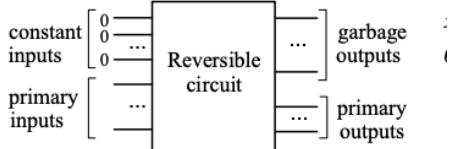


Fig. 13. Reversible Circuit Structure

During the calculation, every operation in the logical expression requires a bit to store the temporal intermediate results in temporary variables. This bit is called *ancillae* bit. The growth of ancillae bits is proportional to the length of the computation. This growth requires space, and without cleaning the ancillae bits, or *collect the garbage*, it will lead to space capacity problems. Since we have reversible gates, we could clean our ancillae by applying the same set of gates in the reverse order, but we would return the entire system's state to the initial state, resetting our desired output. Charles Bennett proposed a method to manage ancillae while retaining the computation's output. In a circuit diagram, an ancillae bit is often called an ancillae line or garbage. The process involves three main steps:

- 1) **Forward Computation:** Perform the computation using reversible gates, storing intermediate results in temporary variables (TEMPVARS).
- 2) **Copy Results:** Copy the results to a dedicated output register (OUTPUT) to preserve them.
- 3) **Reverse Computation:** Reverse the computation to return ancillae to their initial state, effectively "cleaning" them without affecting the final output.

An example reversible circuit of a 5-bit ripple-carry adder with one ancillae (Figure 15) [17] where values of a_i and a_{i+1} are recovered after computation. The MAJ (Majority) and UMA (UnMajority and Add) gates are constructed with NOT and Toffoli Gates (Figure 14).

After the computation of the carry bit c_i , the MAJ gate writes c_{i+1} into A_i (storage notation of a_i). Then with c_{i+1} , the UMA gate restores a_i to A_i and c_i to A_{i-1} and writes s_i to B_i (storage notation of b_i).

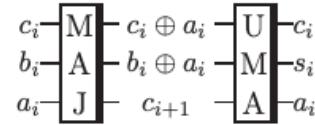


Fig. 14. MAJ with UMA gates

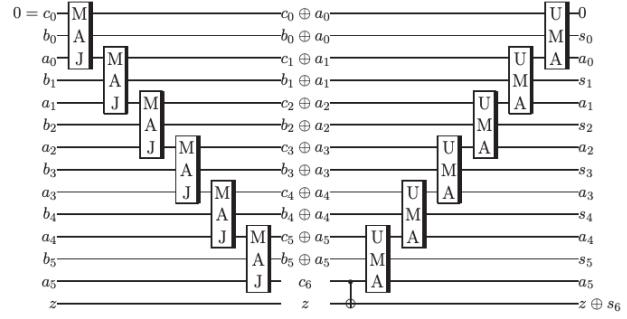


Fig. 15. 5-bit ripple-carry adder

Synthesizing reversible circuits involves transforming a given Boolean function into a reversible form. Recent solutions have enabled the automatic realization of large functions and complex logic as reversible circuits through a hierarchical synthesis approach using Binary Decision Diagrams (BDDs). This method decomposes functions into smaller sub-functions, recombining them to form the desired circuit. A dedicated Hardware Description Language (HDL) ensures reversibility by distinguishing between reversible and non-reversible operations. However, this process often results in numerous additional circuit signals due to non-reversible operations needing extra lines, presenting challenges in optimizing reversible circuit design. To implement an irreversible specification using reversible gates, ancillae (extra bits) should be added to the original specification. The number of added lines, their values, and the ordering of output lines significantly affect the cost of synthesized circuits. This process can either be performed

before synthesis or integrated during synthesis. Saeedi and Markov formalized the process of reversible circuit design and optimization [18].

C. Physical Realization

- **Billiard Ball Model (BBM)**, proposed by Fredkin and Toffoli, is a conceptual model for reversible computing. Instead of using electronic signals, it uses the motion of billiard balls in a friction-free environment, in which the balls bounce perfectly. The balls' trajectories and collisions represent logical operations, with their reversibility ensuring that the computation can be run backward to retrieve the initial state. This model illustrates the physical realization of reversible computing through kinetic systems, highlighting how momentum conservation and elastic collisions can implement logical functions without energy loss [19].

For example, the AND Gate (Figure ??) representation in BBM can be physically realized using the following process. When a single billiard ball enters the gate through either the 0-in or 1-in input, it passes straight through and exits via the corresponding 0-out or 1-out output. However, if balls enter simultaneously through both 0-in and 1-in, they collide at the upper-left corner, redirect to the lower-right corner, and then exit differently: one ball exits via 1-out, and the other exits through the AND-output. Therefore, a ball exiting from the AND-output indicates that balls entered through both 0-in and 1-in, just like the output of an AND gate.

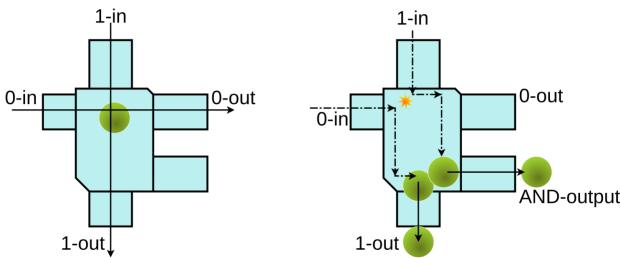


Fig. 16. Billiard Ball Model of an AND Gate

- **Adiabatic CMOS**, uses adiabatic circuits ensuring that energy exchange minimizes loss, reducing heat dissipation associated with traditional CMOS technology. This is an area under research in which (Michale P. Frank et al. 2020) introduced Static 2-Level Adiabatic Logic (S2LAL). A CMOS logic family that is both fully static and fully adiabatic. S2LAL maintains stability by keeping all nodes connected to voltage references and uses 8-phase trapezoidal power-clock wave-forms to minimize energy loss. It achieves low latency (1 tick per logic stage) and high throughput (8-tick clock period) with reduced clock complexity. S2LAL represents a significant step towards energy-efficient reversible computing [20].
- **Quantum Computing**, naturally supports reversible operations due to the unitary nature of quantum mechanics [21].

- **Optical Computing**, leverages reversible light-matter interactions to perform computations. Devices such as all-optical logic gates and optical memories can be designed to operate reversibly.
- **Quantum Cellular Automaton**, extends classical cellular automata to the quantum realm. QCA differs from classical cellular automata in their computational rules and state evolution, which follow unitary transformations to preserve quantum coherence [22].
- **Field Programmable Gate Arrays (FPGAs)** in CMOS technology, for extremely low power [23].

IV. COMPUTING WITH SOLITONS

A. Introduction

A soliton is a strongly stable wave packet that preserves its shape and velocity upon nonlinear interaction with other wave packets. John Scott Russell observed this kind of wave in water waves and first described and reproduced this phenomenon in 1834. Solitons have been explored in many domains, such as Optics (Fiber Optic systems), Biology (motion in proteins and DNA), material physics, and more. Solitons in nonlinear systems, such as optical fibers and photo-refractive crystals, are the most promising candidates to envision in computation. Their stability and robustness under interactions make them ideal candidates for computing applications, particularly with the development of vector solitons, which involve multiple interacting components.

Solitons with nonlinear wave propagation are called envelope solitons. They consist of a carrier wave moving at a characteristic phase velocity, modulated by an envelope moving at a characteristic group velocity (Figure 17).

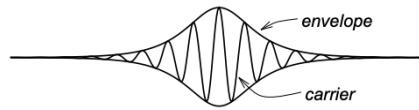


Fig. 17. Envelope Soliton

B. Abstract Realization

When a pulse travels through an optical fiber, it experiences dispersion or temporal spreading. As the pulse propagates, it can be broken down into a spectrum of frequencies; the shorter the pulse duration, the wider its spectral range. The frequency-dependent refractive index causes different frequencies to travel at different speeds, leading to dispersion. In normal dispersion, longer wavelengths travel faster, while in anomalous dispersion, shorter wavelengths travel faster.

The response of dielectrics like optical fiber is nonlinear. If the medium has normal dispersion, the pulse broadens, while in anomalous dispersion, the pulse compresses. Under the proper conditions, the nonlinear response of the fiber, primarily through self-phase modulation (SPM), can counteract dispersion, leading to soliton formation. The nonlinear Schrödinger equation (NLSE) describes this phenomenon, allowing for the

propagation of solitons without distortion. The NLSE, known as scalar NLSE, models the pulse propagation and describes the evolution of complex wave envelopes in a nonlinear medium:

$$i \frac{\partial u}{\partial z} \pm \frac{1}{2} \frac{\partial^2 u}{\partial x^2} + |u|^2 u = 0, \quad (5)$$

Where:

- Complex Wave Envelope $u(z, x)$:
 - u represents the complex envelope of the optical field.
 - z is the propagation distance and x is the propagating time.
 - $|u|^2$ represents the intensity of the wave.
- Linear Evolution Term $i \frac{\partial u}{\partial z}$:
 - i indicates a phase change of the wave.
- Dispersion Term $\pm \frac{1}{2} \frac{\partial^2 u}{\partial x^2}$:
 - Accounts for the dispersion of the wave.
 - + for anomalous dispersion.
 - – for normal dispersion.
- Nonlinear Term $|u|^2 u$:
 - Represents the nonlinear interaction of the wave with the medium.
 - Describes phenomena like self-phase modulation (SPM).

In this model, the wave component described by the Complex Wave Envelope function and solitons are called scalar solitons. During the scalar soliton collision, local intensity increases, resulting in position and phase shifts. These effects are independent of any soliton properties that are changed by the collision, and the result of one collision will not affect the result of subsequent collisions. Thus they are “elastic,” with no change in amplitude or velocity occurs as a result of a collision. Scalar solitons are therefore not useful for complex logic or computing, which depend on multiple, cascaded interactions. For example, consider two solitons traveling in opposite directions within an optical fiber. As they intersect, the overlap results in a higher local intensity, inducing a temporary phase shift and causing each soliton to alter its position slightly. After passing through each other, the solitons retain their original shapes and velocities, thus illustrating the elastic collision (Figure ??) [24].

Unlike scalar solitons, vector solitons involve multiple wave components that interact through nonlinear coupling. This interaction allows for the redistribution of energy between components during collisions, enabling the transfer of information and making them suitable for computational applications. Inelastic collisions (Figure 19) involve changes in the soliton’s properties, and they may generate additional solitons, change their shape, or alter their energy. These interactions can be exploited to create optical logic gates, where the generation of new solitons or changes in existing solitons represent different computational states or outputs. Vector solitons are described by systems of coupled NLSEs, such as the Manakov system, which allows for more complex and versatile behaviors compared to scalar solitons. Therefore, vector solitons are ideal

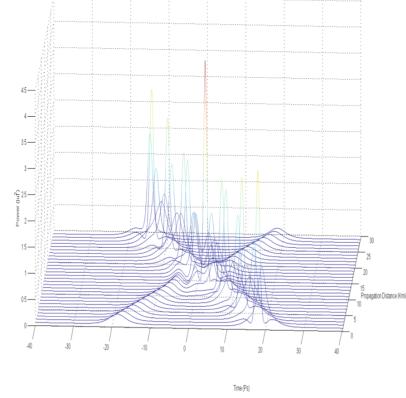


Fig. 18. Elastic collision

for implementing logical operations and other computational functions.

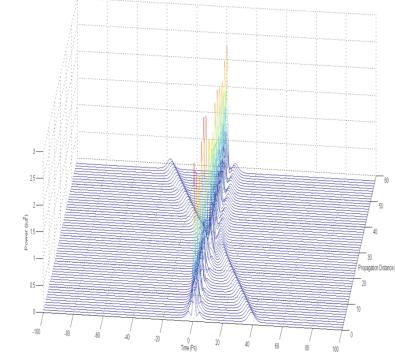


Fig. 19. Inelastic collision

Manakov solitons, a specific type of vector soliton described by coupled NLSEs, exhibit rich collisional dynamics essential for computing. When these solitons collide, their energy redistributes between components, which can be harnessed to perform logical operations. This property is critical for developing a soliton-based computing system that leverages the interactions between solitons to process information. The Manakov system is particularly notable for its integrability and the preservation of soliton properties during collisions, which are crucial for reliable and predictable computational processes.

The Manakov system, which describes the propagation of vector solitons with two interacting components in a nonlinear medium, is given by the following set of coupled nonlinear Schrödinger equations:

$$i \frac{\partial q_1}{\partial z} + \frac{\partial^2 q_1}{\partial x^2} + 2\mu(|q_1|^2 + |q_2|^2)q_1 = 0, \quad (6)$$

$$i \frac{\partial q_2}{\partial z} + \frac{\partial^2 q_2}{\partial x^2} + 2\mu(|q_1|^2 + |q_2|^2)q_2 = 0, \quad (7)$$

where:

- $q_1(x, z)$ and $q_2(x, z)$ are the pulse envelopes of the two interacting optical components.

- μ is a positive parameter representing the strength of the nonlinearity.

In Manakov computing, the state of a soliton is defined with the tuple (ρ, κ) , where:

- $\rho = \frac{q_1}{q_2}$, is the complex number polarization state, as a ratio of the two interacting components
- $\kappa = k_R + ik_I$, is the complex number of the soliton's energy (real part) and soliton's velocity (imaginary part)

As each soliton propagates through the medium, the state remains the same. This means that the soliton's energy and velocity and the relationship between the two components do not change during propagation.

When two solitons collide, their states change due to the interaction. The new states after the collision are determined by specific mathematical transformations that depend on the initial energy, velocity, and polarization states.

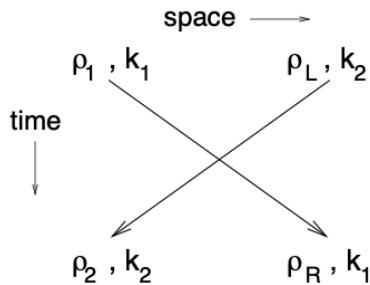


Fig. 20. Manakov System

Consider two wave components, as depicted (Figure 20). The left soliton has the initial state (ρ_1, κ_1) , and after the collision with the right soliton, its state changed to (ρ_R, κ_1) . This proves that the energy and velocity remain constant while the polarization state changes.

The transformation of the state is given by a Linear Fractional Transformation:

$$\rho_R = \frac{[(1 - h^*)/\rho_L^* + \rho_L] \rho_1 + h^* \rho_L / \rho_L^*}{h^* \rho_1 + (1 - h^*) \rho_L + 1/\rho_L^*} \quad (8)$$

$$\text{where } h \equiv \frac{k_2 + k_2^*}{k_1 + k_1^*} \quad (9)$$

The ability to control and manipulate the state of Manakov solitons is fundamental for using them in computing applications due to the following properties:

- **Information Encoding:** The complex polarization state ρ can encode information, with different states representing different bits of data.
- **State Transformation:** The predictable changes in state during soliton collisions allow for the implementation of logical operations and gates, essential for computational processes.
- **Robustness:** The integrable nature of the Manakov system ensures that soliton states are preserved during propagation, making them robust carriers of information over long distances.

Computational logic can be built in a similar method as Reversible Computing. The sequences of soliton collisions that affect a logical operation can act as a logic gate. Similar to digital computers we can have logic gates and memory. A convenient method to represent the usage of solitons in a logical operator has the role of "operator" and the role of "data" (Figure 21).

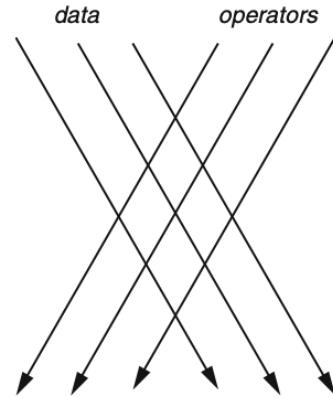


Fig. 21. Convenient representation of colliding spatial soliton

Encoding information in data solitons and operating on them with operator solitons can realize a NOT gate in the following form, as depicted in Figure 20. The pair of the two left-most solitons in both graphs represent one bit of encoded information. Both soliton pairs have opposite phase states ρ between them (1 and -1), to encode the bit of information. In the left graph the pair encodes a bit with value 0 and in the right graph with value 1. After the processing with operator solitons, these bits flipped, resulting to a NOT logic gate (Figure 22).

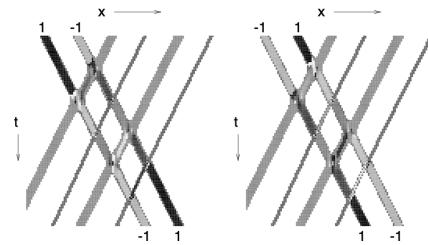


Fig. 22. NOT Gate with Manakov Solitons

Another Soliton computational model proposed in the 3NLSE Domain uses the notion of First-Order and Second-Order Solitons. First-order solitons are simple, stable wave packets that maintain their shape and speed due to a balance between dispersion and nonlinearity in the fiber. Second-order solitons are more complex, exhibiting oscillatory behavior as they propagate. These solitons effectively combine two first-order solitons and can interact with other solitons to produce new solitons or alter existing ones [25].

In this model, the binary digits 0 and 1 are represented by the presence or absence of a soliton or by the position of a soliton relative to another soliton. For example, a soliton detected at a specific position can represent a binary '1', while the absence of a soliton or its presence at a different position can represent a binary '0'.

Figure 23 and 24 showcasing the two optical components and the inputs. In the first component, the first-order soliton formed on the left encodes the bit 0, while in the second optical component, the first-order soliton formed on the right encodes the bit 1. The data and operator solitons, are forming a NAND logic gate, which can be realized by using the OR logic gate on input digits that have initially been inverted (individually passed through a NOT logic gate).

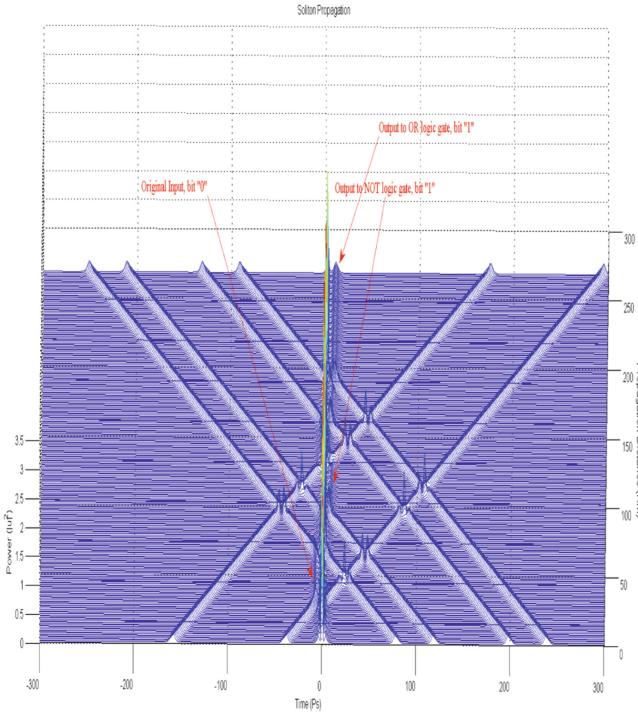


Fig. 23. NAND Logic Gate with 3NLSE, first optical component

C. Physical Realization

- **Optical Fibers**, maintain their shape over long distances, counteracting dispersion. Experiments with fibers have demonstrated stable propagation and collision of temporal vector solitons, making them ideal for computing applications.
- **Photorefractive Crystals**, exhibit strong nonlinear optical properties, supporting the formation of spatial solitons. These materials are used to study soliton interactions and collisions, applicable in image processing and optical information storage.
- **Semiconductor Waveguides**, offer high nonlinearity and are used for integrated photonic circuits. They facilitate the realization of soliton-based logic gates and computing elements, with materials like gallium arsenide (GaAs) and indium phosphide (InP).

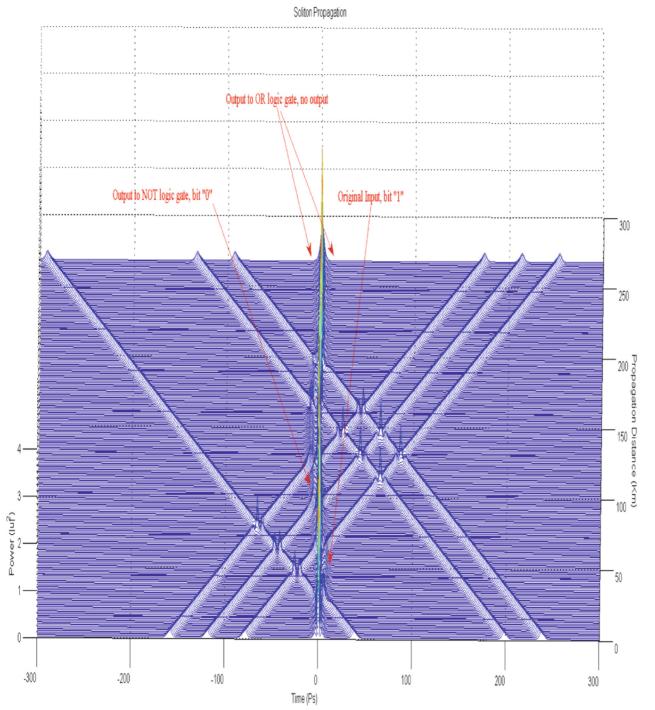


Fig. 24. NAND Logic Gate with 3NLSE, second optical component

- **Bose-Einstein Condensates (BECs)**: BECs provide a macroscopic quantum state where solitons can form and interact. These condensates offer a platform for studying soliton dynamics in a quantum regime, with precision measurements and quantum information processing applications.
- **Solitons in multi-level cellular automata** Soliton-type effects, such as fusion and elastic collisions, are possible for CA with nonlinear evolution rules [26].

V. OPTICAL COMPUTING

A. Introduction

Optical computing uses light waves to perform computations. In this paradigm, photons, the elementary particles of light, serve as the information carriers. Optical computing (OC) began in the 1960s, focusing on Fourier transform applications in optical imaging. Despite their speed, early analog processors struggled with dynamic range and signal-to-noise ratios, leading to a preference for digital electronic processing. Interest in OC remained due to its potential high processing speeds [27].

The 1980s and 1990s saw significant advancements with digital optical computing. In the 1980s, nonlinear optical tools for digital processing functions like AND, OR, and NAND in 2D formats fueled further interest. However, practical challenges like maintaining optical power control and managing complex free-space optics hindered progress. This era introduced parallel logic gates and optical neurocomputing, inspired by neural networks. Despite early potential, the research in OC faded due to immature technologies and advances in electronic computing [28].

Interest revived in the 21st century, driven by Artificial Intelligence and quantum computing needs. Light-based systems offer low-loss transmission, broad bandwidth, and the ability to handle multiple communication streams simultaneously. Optical computing promises to create much faster computers by replacing electrons and wires with photons and optical components. Optical processing is crucial as artificial intelligence (AI) demands grow while Moore's Law slows in silicon-based computing [29].

Optical computing (OC) includes digital optical computing (DOC) and analog optical computing (AOC). DOC, based on Boolean logic like traditional computing, encoding the presence or absence of light to represent bits. AOC uses light's physical properties, such as amplitude and phase, to perform specific tasks. It excels in specialized applications like pattern recognition and numerical calculations, offering faster data processing than standard digital computing. AOC is gaining academic attention as a promising technology for the post-Moore era.

The three primary hardware components of an optical computer are a light source, a modulator, and a detector. The light source generates the necessary light for the system, serving as the initial point of illumination and providing a consistent beam of light that will be manipulated and processed by the subsequent components. The modulator then modifies the light by applying optical functions of computation. Finally, the detector senses the light that has been modified by the modulator. It captures the resulting light and converts it into a signal that can be further analyzed or processed.

There are two main types of optical computer architectures (Figure 25). Processor-Based Architecture uses the traditional electronic computer system structure but replaces the electronic processing unit with an optical processing unit. Parallel Optical Interconnection Network, is substantially different from traditional architectures. It eliminates clear boundaries between processors, buses, and memory, instead using a parallel optical interconnection network for the main operations [27].

B. Abstract Realisation

The Abstract Realisation of the computational model in OC, is very different from other computing paradigms. There are multiple Abstract models, forming various computational models under analog and digital domains. In the literature, these models are classified under Optical Information Processing. The history of Optical Computing lies in the evolution and the diversity of different domains, combined to advance the optical information processing technologies.

Fourier Optical Processing

After the invention of Lasers in the 60s, Holography was realized for Optical Information processing. Holography is a technique for creating 3D images by using the principles of interference and diffraction of light. Discovered by D. Gabor in 1948, it records and reconstructs information about the amplitude (brightness) and the phase (wave position) of the light scattered from an object. In addition, Fraunhofer diffraction

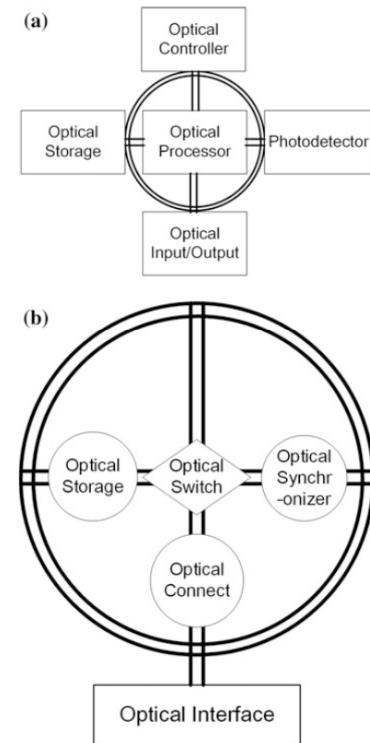


Fig. 25. Optical Computer Architectures

equation to model the diffraction of a wave, observed as 2D Fourier transform, using a lens. Optical Fourier transformation and Holography led to the realization of many Optical Processing techniques in the Analog Computing Domain.

Spatial Filtering systems are using spatial light modulators between Fourier transform of lenses to modulate the light signal with various nonlinear optical phenomena.

The pulse shaping technique uses holographic gratings for the input and output of light, transfers the light with Fourier transform lenses, and modulates it with Spatial Filtering. The optical processing is executed at the speed of light, and along with the fixed phase information of holography, it can achieve high-precision results (Figure 26).

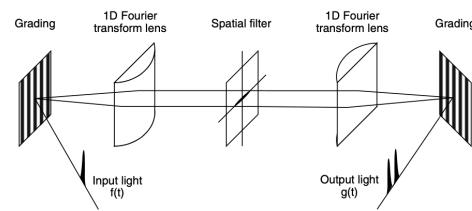


Fig. 26. Pulse shaping technique.

Optical Neurocomputing

Neural networks are well-suited for optical applications due to their need for parallel operation of many similar processing nodes. This is efficiently managed with parallel computing devices and free-space optical interconnection, which removes the need for physical wiring and allows dynamic changes in

connection weights. An optical neural network is composed of an input layer, multiple hidden layers and an output layer. In a complex-valued design, the light signals are encoded and manipulated by both magnitude and phase during the initial input preparation and network evolution (Figure 27).

An early optical neural network system, performed vector-matrix multiplication (VMM) by stretching input signals vertically, projecting them onto a two-dimensional image to represent connection weights, and collecting the output with a one-dimensional image sensor. Recent successes in deep neural networks have led to new optical neuro-processors. The diffractive deep neural network (D2NN) uses stacked diffractive optical elements, while the optical integrated circuit processor employs enhancing complex AI calculations.

In addition, Optical Reservoir Computing derived from recurrent neural network. It achieves performance comparable to recurrent neural networks by only learning the connection weights from the reservoir to output nodes, making it suitable for physical implementation through methods like optical fiber rings, lasers with optical feedback, and optical iterative function systems.

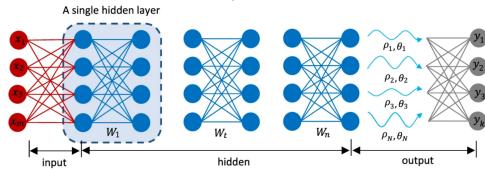


Fig. 27. Optical Neural Network

Digital Optical Computing

Digital optical computing (DOC) up to the 2000s primarily focused on using discrete optics to achieve binary logic, but the sheer number of elements required made it challenging. Despite this, significant results emerged, such as Miller's criteria for practical optical logic. The development of integrated photonics and photonic integrated circuits (PICs) offers a new path to overcome past challenges, necessitating a renewed examination of earlier results. Important topics included the architecture of optical computers, with various proposals in the late 80s and early 90s, such as systolic arrays and crossbars by Sawchuk and Strand and SEED Optical Logic Gates.

In DOC various architectures have been explored. Register-oriented architectures, exemplified by the Tse computer, process data on an image-by-image basis. Interconnection-oriented architectures, such as the OCULAR system and three-dimensional optoelectronic stacked processors, use optical interconnections to link electronic processors (Figure 28) [30]. Although these systems have shown proof-of-principle on a small scale, practical applications remain limited.

C. Physical Realization

- **All-Optical General-Purpose CPU**, introduced efficient all-optical, cross-domain XPU computing architecture for high-performance computing tasks. This architecture features a fully general-purpose CPU capable of operating

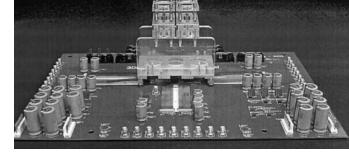


Fig. 28. 3D Optoelectronic Stacked Processor

all-optically, eliminating the need for data conversion to the electronic domain at any point and maintaining all-optical operations throughout each compute cycle. The physical implementation includes key components such as logic and memory, which are integrated into a cohesive system. In this implementation, optical computing can achieve trillions of operations and data transfers per second per core, with capabilities for reversible and cross-domain computing (combining digital, analog, and quantum elements). This positions optical processors to address current challenges in electronic computing [31].

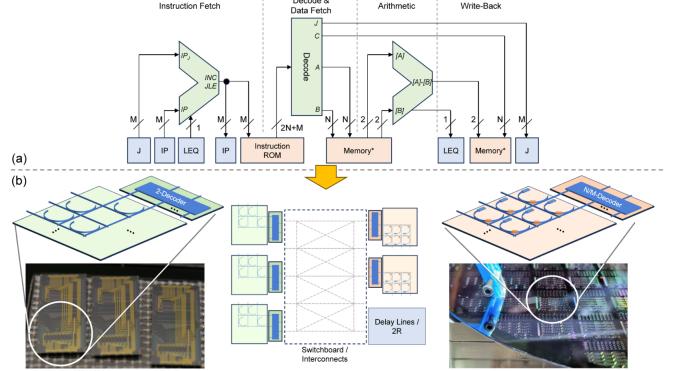


Fig. 29. Implementation of the first all-optical CPU

Figure 29 (a) shows the steps that the processor performs, with green for logic operations, orange for memory access, and blue for registers (a). The M denotes the width of the instruction ROM address space, and N is the width of the memory address space. Figure 29 (b) shows the photonic implementation of (a) with the same color scheme.

- **Optical Neural Chip (ONC) for implementing complex-valued neural network** leverages photonic circuits to perform complex-valued arithmetic efficiently. The chip (Figure 30) consists of integrated photonic elements like waveguides, phase shifters, and optical modulators that manipulate light to represent and process data. By using light's inherent properties, such as phase and amplitude, the chip can perform multiplication and addition operations needed for neural network computations at high speeds and with low power consumption. This architecture is demonstrated in tasks like logic gate realization, species classification, nonlinear dataset classification, and handwriting recognition, showing superior performance and efficiency compared to traditional electronic methods [32].

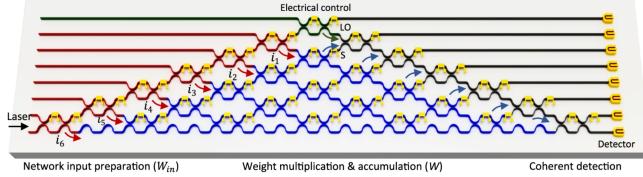


Fig. 30. Schematic Implementation of ONC

- **LightOn Optical Processing Unit**, introduces an innovative photonic AI accelerator designed to enhance AI and high-performance computing (HPC) tasks. The Optical Processing Unit (OPU) utilizes light scattering in disordered media to perform high-dimensional, high-speed, and low-power computations. The photonic approach offers advantages such as energy efficiency, improved robustness to adversarial attacks, and scalable performance, addressing the limitations of conventional electronic processors in handling large-scale AI models and extensive data processing tasks [33].
- **Photonic Quantum Computing**, uses photons as quantum information carriers due to their ability to operate at room temperature and their relatively mature technological infrastructure. The realization of photonic quantum computation can follow two main approaches: discrete-variable (DV) and continuous-variable (CV) photonic quantum computation. In the DV approach, quantum information is encoded in discrete properties of photons, such as polarization, which can be manipulated using optical components like waveplates and beamsplitters, and measured using single-photon detectors. The CV approach, on the other hand, encodes quantum information in continuous properties of the electromagnetic field, such as the quadratures of the electric field, which are processed through Gaussian and non-Gaussian operations and detected via techniques like homodyne detection. Experimental setups often combine both DV and CV methods to leverage the advantages of each, employing advanced technologies such as linear optical elements, single-photon sources, and high-efficiency detectors to create and manipulate the quantum states of light [34]. Photonic quantum computers can operate at room temperature and integrate into existing fiber optic telecommunications infrastructure, facilitating the creation of quantum networks and a potential quantum Internet. On September 2020, Toronto-based Xanadu announced the first publicly available photonic quantum computing platform with 8 (Figure 31), 12, and soon 24 qubit machines over the cloud [35].

VI. CONCLUSION

Unconventional approaches offer promising pathways to overcome CMOS challenges. Each paradigm leverages unique physical principles and materials to enhance computational capabilities and efficiency, but they face critical challenges before large-scale systems are realized and fabricated. In reversible computing, even minor errors can propagate backward, affecting previous states. Soliton and optical computing

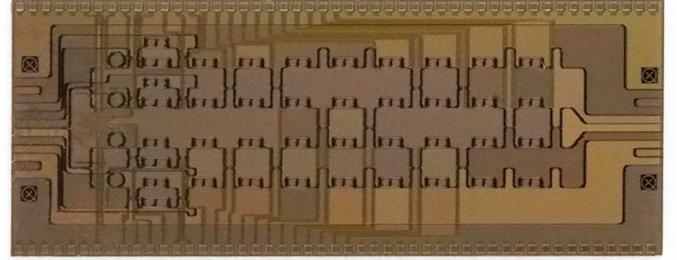


Fig. 31. Xanadu X8 Photonic QPU

also require robust methods to detect and correct errors caused by environmental disturbances or imperfections in materials.

Reversible computing is crucial for low-power and high-efficiency applications like quantum computing, adiabatic CMOS, and nanotechnology. However, technical challenges exist, such as the complexity of designing reversible logic gates and circuits and scaling reversible computing to large, complex systems while maintaining energy efficiency and performance. Additionally, the need for high precision in reversible operations introduces strict requirements for error correction and fault tolerance, making it difficult to maintain accuracy in large-scale applications. The limited availability of materials and technologies that support reversible operations also poses a barrier to wider practical implementation.

Soliton computing offers high-speed signal processing, but generating and manipulating solitons require sophisticated optical setups, which can be sensitive to environmental conditions. The scalability of soliton-based systems is also a concern, as maintaining the stability of multiple interacting solitons in a large-scale network is complex.

Optical computing offers the potential for extremely high-speed and parallel computation. Applications range from advanced artificial intelligence to high-performance computing and cryptography. The major challenges involve developing efficient and reliable optical components, such as modulators, switches, and detectors, managing thermal issues arising from high-intensity optical operations, seamlessly integrating optical systems with conventional electronic computing infrastructure to leverage the strengths of both domains, and scaling up manufacturing processes for optical components to be cost-effective for widespread use. Moreover, the design of optical circuits that can perform complex computations with the same reliability as electronic circuits is a significant complex.

Scaling up production to make these technologies economically viable for commercial use is a significant challenge that requires advancements in fabrication processes and economies of scale. Establishing industry standards like Moore's Law for designing, fabricating, and implementing unconventional computing components is necessary to ensure compatibility and interoperability across different systems and applications. As we advance into the Fourth Industrial Revolution, embracing these unconventional computing paradigms will be essential for meeting the growing demands of modern computational needs.

REFERENCES

- [1] Schwab, Klaus. (2016) The fourth industrial revolution: What it means and how to respond. [Online]. Available: <https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>
- [2] C. Mannarchive. (2000, May) The end of moore's law. [Online]. Available: <https://www.technologyreview.com/2000/05/01/236362/the-end-of-moores-law/>
- [3] H. Huff, "Moore's law beyond planar silicon cmos," in *Into The Nano Era*. Springer Science & Business Media, 2008, vol. 106, ch. 1, pp. 12–14.
- [4] J. Shalf, "The future of computing beyond moore's law," *Philosophical Transactions of the Royal Society A*, vol. 378, no. 2166, p. 20190061, 2020.
- [5] H. Huff, "Moore's law beyond planar silicon cmos," in *Into The Nano Era*. Springer Science & Business Media, 2008, vol. 106, ch. 1, p. 35.
- [6] L. Xiu, "Time moore: Exploiting moore's law from the perspective of time," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 1, pp. 39–55, 2019.
- [7] H. Jaeger and F. Catthoor, "Timescales: the choreography of classical and unconventional computing," 2023. [Online]. Available: <https://arxiv.org/abs/2301.00893>
- [8] Edwards,Lin. (2010, November) Maxwell's demon: Now in the real world. [Online]. Available: <https://phys.org/news/2010-11-maxwell-demon-energy.html>
- [9] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, 1961.
- [10] —, "The physical nature of information," *Physics Letters A*, vol. 217, no. 4, pp. 188–193, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0375960196004537>
- [11] A. Adamatzky, S. Akl, M. Burgin, C. S. Calude, J. F. Costa, M. M. Dehshibi, Y.-P. Gunji, Z. Konkoli, B. MacLennan, B. Marchal, M. Margenstern, G. J. Martínez, R. Mayne, K. Morita, A. Schumann, Y. D. Sergeyev, G. C. Sirakoulis, S. Stepney, K. Svozil, and H. Zenil, "East-west paths to unconventional computing," *Progress in Biophysics and Molecular Biology*, vol. 131, pp. 469–493, 2017, integral Biomathics 2017: The Necessary Conjunction of Western and Eastern Thought Traditions for Exploring the Nature of Mind and Life. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0079610717301177>
- [12] D. Horsman, S. Stepney, R. C. Wagner, and V. Kendon, "When does a physical system compute?" *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 470, no. 2169, Sep. 2014. [Online]. Available: <http://dx.doi.org/10.1098/rspa.2014.0182>
- [13] S. Stepney and V. Kendon, "The representational entity in physical computing," *Natural Computing*, vol. 20, no. 2, pp. 233–242, 2021. [Online]. Available: <https://doi.org/10.1007/s11047-020-09805-3>
- [14] G. Finocchio, J. A. C. Incorvia, J. S. Friedman, Q. Yang, A. Giordano, J. Gröller, H. Yang, F. Ciubotaru, A. V. Chumak, A. J. Naeemi, S. D. Cotofana, R. Tomasellos, C. Panagopoulos, M. Carpentieri, P. Lin, G. Pan, J. J. Yang, A. Todri-Sanial, G. Boschetto, K. Makasheva, V. K. Sangwan, A. R. Trivedi, M. C. Hersam, K. Y. Camsari, P. L. McMahon, S. Datta, B. Koiller, G. H. Aguilar, G. P. Temporão, D. R. Rodrigues, S. Sunada, K. Everschor-Sitte, K. Tatsumura, H. Goto, V. Puliafito, J. Åkerman, H. Takesue, M. D. Ventra, Y. V. Pershin, S. Mukhopadhyay, K. Roy, I. Ting Wang, W. Kang, Y. Zhu, B. K. Kaushik, J. Hasler, S. Ganguly, A. W. Ghosh, W. Levy, V. Roychowdhury, and S. Bandyopadhyay, "Roadmap for unconventional computing with nanotechnology," *Nano Futures*, vol. 8, no. 1, p. 012001, Mar. 2024. [Online]. Available: <http://dx.doi.org/10.1088/2399-1984/ad299a>
- [15] R. D. V. M. I. au2, "Architecture of a quantum multicompiler optimized for shor's factoring algorithm," 2006. [Online]. Available: <https://arxiv.org/abs/quant-ph/0607065>
- [16] P. S. S. C. P.R.Yelekar, "Introduction to reversible logic gates its application," *2nd National Conference on Information and Communication Technology*, vol. NCICT, no. 8, pp. 5–9, November 2011. [Online]. Available: /proceedings/ncict/number8/4237-ncict058/
- [17] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," 2004. [Online]. Available: <https://arxiv.org/abs/quant-ph/0410184>
- [18] M. Saedi and I. L. Markov, "Synthesis and optimization of reversible circuits—a survey," *ACM Computing Surveys*, vol. 45, no. 2, p. 1–34, Feb. 2013. [Online]. Available: <http://dx.doi.org/10.1145/2431211.2431220>
- [19] Wikipedia. (2024) Billiard-ball computer. Accessed: 2024-07-12. [Online]. Available: https://en.wikipedia.org/wiki/Billiard-ball_computer
- [20] M. P. Frank, R. W. Brocato, B. D. Tierney, N. A. Missett, and A. H. Hsia, "Reversible computing with fast, fully static, fully adiabatic cmos," 2020. [Online]. Available: <https://arxiv.org/abs/2009.00448>
- [21] R. Drechsler and R. Wille, "Reversible circuits: Recent accomplishments and future challenges for an emerging technology," in *Progress in VLSI Design and Test*, H. Rahaman, S. Chattopadhyay, and S. Chattopadhyay, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 383–392.
- [22] C. A. Pérez-Delgado and D. Cheung, "Local unitary quantum cellular automata," *Physical Review A*, vol. 76, no. 3, Sep. 2007. [Online]. Available: <http://dx.doi.org/10.1103/PhysRevA.76.032320>
- [23] A. S. M. Sayem and S. K. Mitra, "Efficient approach to design low power reversible logic blocks for field programmable gate arrays," in *2011 IEEE International Conference on Computer Science and Automation Engineering*, vol. 4, 2011, pp. 251–255.
- [24] D. Rand and K. Steiglitz, "Computing with solitons," in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, Ed. New York, NY: Springer New York, 2009, pp. 1376–1395. [Online]. Available: https://doi.org/10.1007/978-0-387-30440-3_92
- [25] A. G. Bakaoukas, "The two separate optical fibres approach in computing with 3nlse-domain optical solitons," in *Intelligent Computing*, K. Arai, R. Bhatia, and S. Kapoor, Eds. Cham: Springer International Publishing, 2019, pp. 253–280.
- [26] A. Ankiewicz and Y. Nagai, "Solitons in multi-level cellular automata," *Chaos, Solitons & Fractals*, vol. 13, no. 6, pp. 1345–1358, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077901001448>
- [27] X. Li, Z. Shao, M. Zhu, and J. Yang, *Fundamentals of Optical Computing Technology*. Springer, 2018.
- [28] J. Tanida, *Revival of Optical Computing*. Singapore: Springer Nature Singapore, 2024, pp. 3–23. [Online]. Available: https://doi.org/10.1007/978-981-99-5072-0_1
- [29] M. M. Mirsalehi, "Optical information processing," in *Encyclopedia of Physical Science and Technology (Third Edition)*, third edition ed., R. A. Meyers, Ed. New York: Academic Press, 2003, pp. 335–369. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B0122274105005251>
- [30] G. Li, D. Huang, E. Yuceturk, P. J. Marchand, S. C. Esener, V. H. Ozguz, and Y. Liu, "Three-dimensional optoelectronic stacked processor by use of free-space optical interconnection and three-dimensional vlsi chip stacks," *Appl. Opt.*, vol. 41, no. 2, pp. 348–360, Jan 2002. [Online]. Available: <https://opg.optica.org/ao/abstract.cfm?URI=ao-41-2-348>
- [31] M. Kissner, L. D. Bino, F. Päslér, P. Caruana, and G. Ghalanos, "An all-optical general-purpose cpu and optical computer architecture," 2024. [Online]. Available: <https://arxiv.org/abs/2403.00045>
- [32] H. Zhang, M. Gu, X. Jiang et al., "An optical neural chip for implementing complex-valued neural network," *Nature Communications*, vol. 12, p. 457, 2021. [Online]. Available: <https://doi.org/10.1038/s41467-020-20719-7>
- [33] C. Brossollet, A. Cappelli, I. Carron, C. Chaintoutis, A. Chatelain, L. Daudet, S. Gigant, D. Hesslow, F. Krzakala, J. Launay, S. Mokaadi, F. Moreau, K. Müller, R. Ohana, G. Pariente, I. Poli, and E. Tommasone, "Lighton optical processing unit: Scaling-up ai and hpc with a non von neumann co-processor," 2021. [Online]. Available: <https://arxiv.org/abs/2107.11814>
- [34] J. Romero and G. Milburn, "Photonic quantum computing," 2024. [Online]. Available: <https://arxiv.org/abs/2404.03367>
- [35] C. Choi. (2020, Sep) First photonic quantum computer on the cloud. [Online]. Available: <https://spectrum.ieee.org/photonic-quantum>