CHAPTER 1

Introduction

Understanding natural phenomena is one of the biggest challenges for the research community. Nature consists of a set of interacting processes that are responsible for ensuring the balance of its constituents and proper functioning of the system as a whole. A simple example is the balancing of *carbon dioxide* and *oxygen* in the atmosphere through the twin processes of *photosynthesis* and *respiration*. While *photosynthesis* absorbs carbon dioxide and emits oxygen, *respiration* does just the reverse. In a similar fashion, most of the natural processes are bound to each other by a set of *cause-effect* relationships, which are often cyclic in structure. Some of the prominent examples include the *food cycle* in the animal kingdom, the process of *reproduction and growth*, the *heating of earth due to sun rays and rain*, and so on. The *food cycle* defines a hierarchy among the animals whereby the animals at lower levels are eaten by those higher up in the hierarchy; and finally, when an animal of higher class dies, its decomposed body serves as food for the lower classes. Thus, the apparently simple case of distinct food habits of the animals serves to keep the balance in the ecosystem—which cannot be understood adequately unless all the processes in the cycle are considered in an integrated manner. Similar is the case for the *growth* and *reproduction* processes—both of these are responsible for sustenance of life in the universe. Also, when the earth gets heated with sun rays, water evaporates, creating clouds, which in turn cause rain to cycle the evaporated water back to the earth.

In general, a set of apparently simple processes constitute a physical system that poses a mystery when considered as a whole. The most important observation is that it is the interactions among such simple processes that make the whole system exhibit behavior that is too complex to be conceived of only by considering each of the processes in isolation. Further, a single process may be governed by simple rules having limited capability, thus maintaining the homogeneity of the overall structure in terms of composition and degree of complexity of constituent processes of the system. However, the interactions among these simple and homogeneous processes contribute to the complexity and the associated system response. To establish the point further, the glaring example of the neural system of the animal world can be considered. Each neuron cell is simple and limited by its capacity. But their interactions make the system exhibit complex behavior that is in no way comparable to the capabilities of individual neurons. A similar analogy may also be extended to the illustrations considered earlier. Each of the processes of *photosynthesis* and *respiration* can be abstracted as a simple process capable of performing two primitive operations—absorbing a gas and emitting another. The interaction between the two processes maintains the balance in the atmosphere. If each animal is considered as a distinct process with "food habit" as the primitive operation, the *food cycle* gets modeled by another homogeneous system. Thus, nature can be viewed as the collection of all such interacting homogeneous structures.

In the background of the above scenario, the study of *homogeneous structures* has long drawn the attention of researchers. Such a structure consists of a set of cells, each one capable of performing only a set of primitive operations. Depending on the interconnection pattern among the cells and the initial content, the structure evolves through a sequence of states. The study of the homogeneous structure of cellular automata (CA) started with J. von Neumann [1]. In his lectures on self-reproduction, von Neumann constructed a model different from a kinematic model—*the cellular model of self-reproduction*. Each cell in the structure is complex in nature with considerable processing power. The next important milestone in the history of the development of the homogeneous structure of cellular automata is due to Wolfram [2]. He suggested simplification of the cell structure with local interconnections. The CA structure he proposed consists of cells, each having only two states with uniform three-neighborhood interconnections. The function determining the state transitions of each cell is referred to as the *next-state function*. Simplifications introduced by Wolfram made the CA with a linear next-state function amenable for polynomial algebraic analysis similar to that of *linear feedback shift register (LFSR)* [3]. However, the technique cannot be used to analyze CA with nonuniform interconnection structures. Later, Das et al. [4] proposed a versatile matrix algebraic tool for analysis of the state-transition behavior of CA with a linear next-state function. It can handle the CA with nonuniform interconnections also. This in turn motivated researchers to undertake further in-depth studies of CA behavior and look for innovative applications of this linear machine.

The VLSI era has ushered a new phase of activities into the research of linear machines, and especially the local-neighborhood CA structures. The VLSI design community always prefer to have simple, regular, modular, and cascadable structure with local interconnections. With the advancement of semiconductor technology, circuit delay due to interconnections on the silicon floor has become a major concern. Further, in the next-generation submicron technology, interconnections will behave more like a device on the silicon floor (rather than just a simple signal path between the functional modules), thereby contributing a lion's share to the circuit delay. For example, a 12 mm metal line in 0.5 technology, as noted in [5], has twice the delay of logic gates. This situation invariably forces the designers to have local interconnections as far as possible, for reliable high-speed operations of the circuit. The simple, regular, modular, and cascadable structure of CA with local interconnections are ideally suited for VLSI implementation. The class of linear circuits built around LFSRs can be found to have the following inherent disadvantages: (i) irregularity of the interconnection structure, (ii) longer delay, and (iii) lack of modularity and cascadability. So, in this submicron era an alternative structure to LFSR is often sought by the VLSI design community. From the study of its state-transition behavior, it can be observed that CA does provide a cost-effective alternative to the existing linear structures like LFSR. Another advantage of CA over LFSR stems from its wide variety of state-transition behaviors. The state-transition diagram of an LFSR consists of a set of cycles. By contrast, apart from the cyclic state transitions, one can also have the tree-like state-transition behavior in a CA. Further, a new class of machines can also be constructed by inverting the linear next-state function—that is, employing XNOR logic for some (or all) of the cells. The associated change in the state-transition behavior can be analyzed by considering the corresponding linear machine. CA employing only XOR logic as the next-state function are referred to as *linear* CA, whereas CA employing both XOR/XNOR logic are referred to as *additive* CA. In the above context, the next section introduces the major CA applications undertaken by researchers in this field.

1.1 CELLULAR AUTOMATA APPLICATIONS

J. von Neumann [1] framed CA as a cellular space with self-reproducing configurations involving 5-neighborhood cells, each having 29 states. Subsequently, theory and applications of CA have evolved in diverse areas such as pattern recognition, modeling biological and physical systems, parallel computation, and so forth.

Study of the regular structure of CA has drawn considerable attention in recent years. Statistical mechanics of local-neighborhood, one-dimensional CA was first reported by Wolfram in his classic paper [2]. Martin et al. [6] characterized one-dimensional CA using polynomial algebraic formulation. Recently, a more versatile tool based on matrix algebra has been proposed by Das et al. [4, 7] to characterize additive CA behavior.

In the last two decades a wide variety of applications has been proposed. Major applications can be categorized under the following broad headings:

1. Simulation of physical systems—A few notable applications include modeling of growth processes [8, 9], reaction-diffusion systems [10, 11], hydrodynamics [12], and soliton-like behavior [13].
2. Biological modeling involving models for self-reproduction [14, 15, 16], biological structures and processes [17, 18, 19], DNA sequences [20].
3. Image processing [21, 22, 23].
4. Language recognition [24, 25, 26].
5. Computations such as sorting [27], generation of primes [28].
6. Simulation machines [29, 30].
7. Computer architectures [31, 32].

8. BIST (Built-In-Self-Test) structure for pseudorandom, pseudoexhaustive, and deterministic pattern generation [33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43] and signature analysis [44, 38, 45, 46].
9. Synthesis of easily testable finite-state machine (FSM) [47, 48, 49, 50].
10. Error correcting codes [51, 52, 53].
11. Pseudoassociative memory [54, 55].
12. General and perfect hashing function generator [56].
13. Board-level fault diagnosis [57].
14. Mod-p multiplier [58, 59].
15. Block and stream cipher cryptography [60].
16. Fractals and chaos [61, 62, 63, 64, 65].

1.2 OVERVIEW OF THE BOOK

This section provides an overview of the book's remaining chapters.

Chapter 2 provides a comprehensive survey of different phases of developments of cellular automata theory and its applications. The discussions have been centered around four major phases: *initial phase of development, CA models for different applications, new phase of activities,* and finally, the *consolidation in the VLSI era.*

Two-states-per-cell, local-neighborhood additive CA are the focus of this book. On the basis of the properties of state-transition graphs, additive CA can be broadly classified into two classes—*group CA* and *nongroup CA*. While in the state-transition graph of a *group CA* all states belong to some disjoint set of cycles, the nongroup CA are characterized by the presence of some nonreachable states in the state-transition graph. Chapter 3 presents a detailed analysis of the cyclic behavior of *group CA.* The *nongroup CA* are discussed in Chapter 4. Apart from characterizing the cyclic behavior, this chapter also analyzes the tree-structured transition behavior of acyclic states of nongroup CA. The theoretical foundation established in Chapters 3 and 4 are then explored in subsequent chapters to build up a few end-applications of general interest. Chapter 5 establishes CA as a universal pattern generator. Different types of patterns—*pseudorandom, pseudoexhaustive,* and *deterministic* can be generated efficiently with CA. Generation of each of these types of patterns have been discussed in detail. While establishing CA as a universal pattern generator, specific emphasis has been directed for on-chip generation of various test patterns for testing a VLSI chip. Design of efficient *error correcting codes* is another interesting application area. The CA-based design of bit and byte error correcting codes is presented in Chapter 6. The regular, modular, and cascadable structure of CA with local interconnection offers some definite advantages (over the other available schemes) for the design of the decoders. Data encryption is another important area in which CA have been applied successfully to achieve a low-cost solution of the security problem. Chapter 7 deals with the CA-based data encryption technique. Both *stream cipher* and *block cipher* cryptography are addressed. A special class of nongroup CA can also be utilized to generate efficient hashing functions. This application is discussed in Chapter 8. The issues related to both the *general* and *perfect* hashing are presented. The CA-based hashing method performs at par with the existing hashing techniques as far as the *collision probability* is concerned; however, the scheme is much faster than other hashing techniques. One of the most important issues in today's VLSI circuit design is the *synthesis for testability*. Testable synthesis of both *sequential* and *combinational* logic, with CA as the underlying test machine, is elaborated in Chapter 9. All of these applications utilize CA with cells arranged linearly in one dimension. This is extended in Chapter 10 to analyze the behavior of two-dimensional cellular automata as well as its applications in designing a low-cost *pseudoassociative memory*.