Preface

In 1946, Electronic Numerical Integrator and Calculator (**ENIAC**) was the first general-purpose, totally electronic computer constructed by John Mauchly and J. Presper Eckert. It contained 18,000 vacuum tubes and was 10 feet high by 100 feet long with 30 tons weight. John von Neumann offered that the program and the data should be stored in memory. Based on von Neumann's ideas, in 1950 at the University of Pennsylvania the first electronic computer that was called EDVAC was constructed. Electronic computers made after 1950 actually follow von Neumann's ideas and vacuum tubes were replaced by the **integrated circuit** which was transistors, wiring, and other components on a single chip.

In 1959, Richard Feynman (Feynman 1961) gave a visionary talk describing the possibility of building computers which were "*sub*-microscopic." In 1994, with Adleman's seminal work (Adleman 1994) the possibility of computing directly with molecules is explored. Since then significant theoretical and experimental results have been reported regularly for the field of molecular computing. Several famous books (Amos 2005; Calude and Paun 2001; Paun et al. 1998) have introduced various paradigms of molecular computing.

A bit which is either 0 or 1 is the smallest unit of data to solve any problem with the input of n bits and 2^n combination states of n bits are all of the possible solutions for that problem. Molecular algorithms are to make use of bit operations to simultaneously operate 2^n combination states and to obtain the required answer(s). The purpose of the book is to teach a reader to how to design various molecular algorithms to construct various kinds of digital arithmetical and logical circuits. Here is an outline of the chapters:

Chapter 1 explains the behaviors of a digital computer and molecular computing. Simultaneously, a concise description to a digital computer and molecular computing based on the von Neumann Architecture is also given.

Chapter 2 introduces a bit pattern which is a uniform representation of data and is also the most efficient solution to how to handle all of the data types. A hexadecimal system with its base that is 16 and an octal system with its base that is eight are concisely illustrated, and an introduction for the conversion between a hexadecimal number or an octal number and a binary number is also given.

Chapter 3 illustrates how eight molecular operations work and use them to develop a parity counter of n bits, the parity generator of error-detection codes on digital communication, and the parity checker of error-detection codes on digital communication.

Chapter 4 describes the conversion between a decimal number and a binary number and develops molecular algorithms to construct the range of the value for an unsigned integer of n bits, a sign-and-magnitude integer of n bits, a one's complement integer of n bits, a two's complement integer of n bits, a floating-point number of n bits in form of single precision format based on Excess_127, and a floating-point number of n bits in the form of double precision format based on Excess_1023.

Chapter 5 develops molecular algorithms to implement a parallel adder of one bit and a parallel adder of n Bits to unsigned integers, and a parallel subtractor of one bit and a parallel subtractor of n Bits for unsigned integers.

Chapter 6 develops molecular algorithms to construct the parallel **NOT** operation of a bit and n bits, the parallel **OR** operation of a bit and n bits, the parallel **AND** operation of a bit and n bits, the parallel **NOR** operation of a bit and n bits, the parallel **NAND** operation of a bit and n bits, the parallel **XOR** operation of a bit and n bits, and the parallel **XNOR** operation of a bit and n bits.

Chapter 7 develops molecular algorithms to complete parallel comparators of a bit and n bits, a parallel left shifter of n bits, a parallel right shifter of n bits, the parallel operation of increase with n bits, the parallel operation of decrease with n bits, and finding the maximum and minimum numbers of one from 2^n combinations of n bits.

The book consists of extensive exercises at the end of each chapter. Solutions of exercises from Chap. 1 through Chap. 7 can easily be completed by readers if they fully understand the contents of each chapter. Instructors, researchers, and students can find the solutions of all the exercises on http://extras.springer.com/.

Power point presentations have been developed for this text as an invaluable tool for learning. Instructors, researchers, and students can find the Power point presentations of all chapters on http://extras.springer.com/.

October 2013

Weng-Long Chang
Athanasios V. Vasilakos

References

L.M. Adleman, Molecular computation of solutions to combinatorial problems. Sci. **226**, 1021–1024 (1994)

M. Amos, *Theoretical and Experimental DNA Computation*, (Springer-Verlag, 2005), ISBN: 9783540657736

C.S. Calude, G. Paun, *Computing with Cells and Atoms*: *An Introduction to Quantum*, *DNA and Membrane Computing*, (Taylor and Francis, London, 2001), ISBN: 0748408991

R.P. Feynman, in *Miniaturization* (Reinhold Publishing Corporation, New York, 1961), pp. 282–296

G. Paun, G. Rozenberg, A. Salomaa, *DNA Computing: New Computing Paradigms*, (Springer-Verlag, 1998), ISBN: 3540641963

http://www.springer.com/978-3-319-05121-5

Molecular Computing
Towards a Novel Computing Architecture for Complex Problem Solving
Chang, W.-L.; Vasilakos, A.V.
2014, XIII, 271 p. 229 illus., Hardcover
ISBN: 978-3-319-05121-5