Infochemistry
Information Processing at the Nanoscale

KONRAD SZACIŁOWSKI, Faculty of Non-Ferrous Metals, 
AGH University of Science and Technology, Kraków, Poland 
and Faculty of Chemistry, Jagiellonian University, Kraków, Poland

Infochemistry, the science at the interface between chemistry and the information sciences, follows the potential use of molecular switches in advanced information processing and storage. Where traditional transistors are reaching their limits in miniaturization due to interconnectivity and material issues, molecular information processing could be the key to satisfying our ongoing need for increasing computational power.

This book describes the processes, systems and devices employed in infochemistry, and characterizes chemical systems according to their role as potential elements in information technology. It includes:

- Theory of Information
- Limitations of traditional devices
- Low-Dimensional Metals and Semiconductors
- Carbon and Carbon-Rich Nanostructures
- Photocurrent switching and related phenomena
- Supramolecular assembly
- Molecular scale electronics, logic gates and computing systems
- Bioinspired and biomimetic logic devices

Presenting a unique combination of chemistry and information technology, Infochemistry: Information Processing at the Nanoscale defines this new field of science, and its future possibilities for advanced information processing.
Infochemistry
Infochemistry

Information Processing
at the Nanoscale

Konrad Szaciłowski

Faculty of Non-Ferrous Metals, AGH University of Science and Technology, Kraków, Poland and Faculty of Chemistry, Jagiellonian University, Kraków, Poland
For Bela, Maria and Marek, with love
## Contents

**Preface** xi  
**Acknowledgements** xiii  

### 1 Introduction to the Theory of Information 1

1.1 Introduction 1  
1.2 Definition and Properties of Information 2  
1.3 Principles of Boolean Algebra 4  
1.4 Digital Information Processing and Logic Gates 7  
  1.4.1 Simple Logic Gates 7  
  1.4.2 Concatenated Logic Circuits 10  
  1.4.3 Sequential Logic Circuits 11  
1.5 Ternary and Higher Logic Calculi 14  
1.6 Irreversible vs Reversible Logic 16  
1.7 Quantum Logic 18  
References 20  

### 2 Physical and Technological Limits of Classical Electronics 23

2.1 Introduction 23  
2.2 Fundamental Limitations of Information Processing 24  
2.3 Technological Limits of Miniaturization 27  
References 34  

### 3 Changing the Paradigm: Towards Computation with Molecules 37

References 53  

### 4 Low-Dimensional Metals and Semiconductors 63

4.1 Dimensionality and Morphology of Nanostructures 63  
4.2 Electrical and Optical Properties of Nanoobjects and Nanostructures 70  
  4.2.1 Metals 70  
  4.2.2 Semiconductors 84  
4.3 Molecular Scale Engineering of Semiconducting Surfaces 96  
  4.3.1 Semiconductor–Molecule Interactions 100  
  4.3.2 Electronic Coupling between Semiconducting Surfaces and Adsorbates 103  
References 109
# Contents

5 Carbon Nanostructures 119
  5.1 Nanoforms of Carbon 119
  5.2 Electronic Structure and Properties of Graphene 120
  5.3 Carbon Nanotubes 129
  5.4 Conjugated and Polyaromatic Systems 139
  5.5 Nanocarbon and Organic Semiconductor Devices 149

References 156

6 Photoelectrochemical Photocurrent Switching and Related Phenomena 165
  6.1 Photocurrent Generation and Switching in Neat Semiconductors 165
  6.2 Photocurrent Switching in MIM Organic Devices 168
  6.3 Photocurrent Switching in Semiconducting Composites 178
  6.4 Photocurrent Switching in Surface-Modified Semiconductors 181

References 192

7 Self-Organization and Self-Assembly in Supramolecular Systems 199
  7.1 Supramolecular Assembly: Towards Molecular Devices 199
  7.2 Self-Assembled Semiconducting Structures 201
  7.3 Self-Assembly at Solid Interfaces 210
  7.4 Controlling Self-Assembly of Nanoparticles 212
  7.5 Self-Assembly and Molecular Electronics 215

References 219

8 Molecular-Scale Electronics 225
  8.1 Electron Transfer and Molecular Junctions 225
  8.2 Nanoscale Electromagnetism 232
  8.3 Molecular Rectifiers 238

References 246

9 Molecular Logic Gates 249
  9.1 Introduction 249
  9.2 Chemically Driven Logic Gates 249
    9.2.1 OR Gates 252
    9.2.2 AND Gates 255
    9.2.3 XOR Gates 267
    9.2.4 INH Gates 272
    9.2.5 IMP Gates 281
    9.2.6 Inverted Logic Gates (NOR, NAND, XNOR) 283
    9.2.7 Behind Classical Boolean Scheme-Ternary Logic and Feynman Gate 289
  9.3 All-Optical Logic Gates 298
  9.4 Electrochemical Logic Systems 307

References 315
**10 Molecular Computing Systems**  
10.1 Introduction 323  
10.2 Reconfigurable and Superimposed Molecular Logic Devices 323  
10.3 Concatenated Chemical Logic Systems 337  
10.4 Molecular-Scale Digital Communication 353  
  10.4.1 Multiplexers and Demultiplexers 354  
  10.4.2 Encoders and Decoders 355  
  10.4.3 Molecular-Scale Signal Amplification 359  
10.5 Molecular Arithmetics: Adders and Subtractors 363  
  10.5.1 Molecular-Scale Half-Adders 363  
  10.5.2 Molecular-Scale Half-Subtractors 372  
  10.5.3 Half-Adders/Half-Subtractors 381  
  10.5.4 Full Adders and Full Subtractors: Towards Molecular Processors 382  
10.6 Molecular-Scale Security Systems 386  
10.7 Noise and Error Propagation in Concatenated Systems 396

References 398

**11 Bioinspired and Biomimetic Logic Devices**  
11.1 Information Processing in Natural Systems 405  
11.2 Protein-Based Digital Systems 408  
  11.2.1 Enzymes as Information Processing Molecules 409  
  11.2.2 Enzymes as Information Carriers 428  
11.3 Binary Logic Devices based on Nucleic Acids 430  
11.4 Logic Devices Based on Whole Organisms 445

References 450

**12 Concluding Remarks and Future Prospects**  

References 458

*Index* 461
Preface:

Why “Infochemistry”?

‘It is a very sad thing that nowadays there is so little useless information.’

Oscar Wilde

For many people, information processing using molecules seems a kind of science fiction. It is hard to imagine fancy netbooks, palmtops and other smart electronic gadgets replaced by jars of snot-like liquid or other *gebuzina*. On the other hand each of us carry the most powerful information processing “device” that can be found anywhere: the brain. At this moment and in any foreseeable future, mimicking our brains with any artificial systems of any kind seems impossible. However, we should keep trying to force molecular systems to compute. While we will not be able to build powerful systems, all this effort can serendipitously yield some other valuable results and technologies. And even if not, the combination of chemistry and information theory paves an exciting path to follow. There is a quote attributed to Richard P. Feyman saying: “Physics is like sex. Sure, it may give some practical results, but that’s not why we do it”. With infochemistry is it exactly the same!

When approximately half of the manuscript was ready I realized that it was going to be almost a “useless” book. For most chemists it may be hard to follow due to the large amount of electronics content, while for electronic engineers there is far too much chemistry in it. And both fields, along with solid-state physics, are treated rather superficially, but are spiced with a handful of heavy mathematics and a couple of buckets of weird organic structures. But then I found that rather optimistic sentence by Oscar Wilde which motivated me to complete this work.

This book treats the interface between chemistry and information sciences. There are other books which can be located in this field, including my favourites *Ideas of Quantum Chemistry* by Lucjan Piela and *Information Theory of Molecular Systems* by Roman F. Nalewajski. In my book I have tried to show how diverse properties of chemical systems can be used for the implementation of Boolean algebraic operations. The book can be divided into three main sections. The first section (Chapters 1–3) explores the basic

---

1 (Slovak) unidentified disgusting semi-liquid substance of unpleasant smell, swill
principles of the theory of information, the physical and technological limits of semiconductor-based electronics and some alternative approaches to digital computing. The next section (Chapters 4–8) is intended to show how the properties of materials commonly used in classical electronics are modified at the nanoscale, what happens at the molecule/semiconductor interface and how these phenomena can be used for information processing purposes. Finally, the last section (Chapters 9–11) are (I hope) a comprehensive review of almost all molecular logic systems described in the chemical literature from 1993 (the seminal Nature paper by Amilra Prasanna de Silva) to November 2011, when my “Хождение по мукам”\textsuperscript{2} was over.

\textsuperscript{2} An original title of a novel by Aleksey Nikolayevich Tolstoy. The title was translated as “The Road to Calvary”, but its literal meaning is rather “walking through torments”.
Acknowledgements

I would like to thank my wife Bela and kids Maria and Marek for their patience, help and support during the preparation of this manuscript. Without their love and understanding this book could not have been written.

I would also like to express my gratitude to my teachers and mentors for their efforts and devotion. First of all I should mention my grandfather Stefan Polus, who showed me the wonderful world of electronics, my PhD supervisor Professor Zofia Stasicka, who introduced me to the realm of inorganic photochemistry and my postdoctoral mentor, Professor John F. Endicott who taught me careful data analysis and skepticism. Large parts of this book were written at The Faculty of Non-Ferrous Metals, AGH University of Science and Technology. Therefore I address my thanks to the Dean of the Faculty, Professor Krzysztof Fitzner for his patience and support.

This book could not have been written without financial support. Most of the manuscript was prepared with support from AGH-UST within contract No. 11.11.180.509/11. Many results presented in this book were obtained within several research projects funded by The Polish Ministry of Science and Higher Education (grants Nos. 1609/B/H03/2009/36, 0117/B/H03/2010/38 and PB1283/T09/2005/29), The National Centre for Research and Development (grant No. NCBiR/ENIAC-2009-1/1/2010), The European Nanoelectronics Initiative Advisory Council JU ENIAC (Project MERCURE, contract No. 120122) and The European Regional Development Fund under the Innovative Economy Operational Programme (grant No. 01.01.02-00-015/09-00), both at The Faculty of Chemistry, Jagiellonian University and The Faculty of Non-Ferrous Metals, AGH University of Science and Technology.

Last but not least I would like to thank my copy-editor Jo Tyszka and the Wiley editorial and production team: Rebecca Stubbs, Emma Strickland, Sarah Tilley, Richard Davies, Tanushree Mathur and Abhishan Sharma.

Thank you!
1

Introduction to the Theory of Information

‘Do not worry about your difficulties in Mathematics. I can assure you mine are still greater.’

Albert Einstein

1.1 Introduction

Information processing is the most important and the most energy consuming human activity. Our brains contain approximately $3 \times 10^9$ neurons and each of them has approximately $10^4$ connections with other neurons. This impressive network is dense, as each cubic millimetre of neural tissue contains up to $10^9$ synaptic junctions. While the brain constitutes only 2% of body mass, it consumes 20% of the energetic demand at rest. We really must need our personal ‘CPUs’ as Mother Nature invests so much energy in nervous systems. The importance of information storage, transfer and processing has been greatly appreciated through the ages. Nowadays various techniques have revolutionized all aspects of information processing via digital electronic technologies.

It is impossible to find a direct relation between brains and computers, but both systems show some functional and structural analogies. Their building blocks are relatively simple and operate according to well-defined rules, the complex functions they can perform is a result of the structural complexity (i.e. is an emergent feature of the system) and communication between structural elements is digital. This is quite obvious for electronic computers, but spikes of action potential can also be regarded as digital.
signals, as it is not the amplitude of the signal, but the sequence of otherwise identical pulses that carries information.

1.2 Definition and Properties of Information

We all intuitively use and understand the notion of information, but it defies precise definition. The concept of information has many meanings, depending on the context. It is usually associated with language, data, knowledge or perception, but in thermodynamics it is a notion closely related to entropy. Its technical definition is usually understood to be an ordered sequence of symbols. Information can be also regarded as any kind of sensory input for humans, animals, plants and artificial devices. It should carry a pattern that influences the interaction of the system with other sensory inputs or other patterns. This definition separates information from consciousness, as interaction with patterns (or pattern circulation) can take place in unanimated systems as well.

While the psychological definition of information is ambiguous, the technological applications must be based on strict definitions and measures. Information can be regarded as a certain physical or structural feature of any system. It can be understood as a degree of order of any physical system. This (structural) form of information is usually regarded as a third (along with matter and energy) component of the Universe. Every object, phenomenon or process can be described in terms of matter (type and number of particles), energy (physical movements) and information (structure). In other words, information can be another manifestation of a primary element. In the same way that the special theory of relativity expresses the equivalence of mass and energy (1.1) [1],

\[ E = mc^2, \]  

the equivalence of energy and information can be shown within information theory (vide infra).

The most precise definition of information is given by the syntactic theory of Hartley and Shannon. According to this theory, information is a measure of the probability of a certain event. It is the amount of uncertainty removed on occurrence of an event or data transmission. The less probable the event, the higher its information value. According to Hartley the amount of information \( (I_i) \) given by an event \( x_i \) can be formulated as (1.2) [2,3]:

\[ I_i = \log_r \frac{1}{p_i}, \]

where \( p_i \) denotes the probability of an event \( x_i \) and \( r \) is the base of logarithm. Such an expressed amount of information is also a measure of the entropy associated with the event \( x_i \). The average amount of information carried by an event from a set of events \( (X) \) is the weighted average of entropies of all the events within this set (1.3):

\[ H(X) = \sum_{i=1}^{n} p_i \log_r \frac{1}{p_i} = -\sum_{i=1}^{n} p_i \log_r p_i \]  

(1.3)
This definition automatically defines the unit of information. Depending on the logarithm base the basic information units are bit \((r = 2)\), nit \((r = e)\) and dit \((r = 10)\). With \(r = 2\) the information content of the event is measured as the number of binary digits necessary to describe it, provided there is no redundancy in the message.

The average amount of information in the system is related to the system entropy by (1.4):

\[
S = -k_B \sum_{i=1}^{n} p_i \log p_i, \tag{1.4}
\]

where \(k_B\) is the Boltzmann constant. As derived by Landauer, the energetic equivalent binary transition of one bit at \(T = 298\) K amounts to (1.5) [4]:

\[
E = k_B T \ln 2 \approx 2.8 \times 10^{-21} \text{ J \cdot bit}^{-1} \tag{1.5}
\]

More precisely, this is the minimum amount of energy that must be dissipated on erasure of one bit of information. This calculation, initially based on the *Second Law of Thermodynamics* was later generalized on the basis of the *Fokker–Planck equation* concerning the simplest memory model and a Brownian motion particle in a potential well [5]. The same value has also been derived microscopically without direct reference to the Second Law of Thermodynamics for classical systems with continuous space and time and with discrete space and time, and for a quantum system [6]. Interestingly, exactly the same value is obtained as the energetic limit required for switching of a single binary switch [7]. Any physical system which can be regarded as a binary switch must exist in two stable, equienergetic states separated by an energy barrier (Figure 1.1a). The barrier must be high enough to prevent thermal equilibration of these two distinct states. At any temperature, mechanical vibration of atoms and the thermal electromagnetic field may induce spontaneous switching between the states.

![Figure 1.1](image)

*Figure 1.1* Energy diagram of a binary switch (a) (Adapted from [5] Copyright (1995) American Physical Society) and schematic representation of a physical system corresponding to a binary switch immersed in a thermal bath with two components: radiation and atomic vibrations (b). (Adapted from [7] Copyright (2006) Springer Science + Business Media.)
The probability of this spontaneous process can be quantified by the error probability \( \Pi_{err} \) obtained from the Boltzmann distribution (1.6) [7]:

\[
\Pi_{err} = \exp\left(-\frac{\Delta E_b}{k_BT}\right)
\]

(1.6)

The minimum barrier height (below this value the two states are indistinguishable) can be calculated assuming the error probability is equal to 0.5. The value of the limiting energy thus obtained for a binary switching process is identical to the energetic equivalent of one bit of information (1.7).

\[
\Delta E_{b,\text{min}} = k_B T \ln 2
\]

(1.7)

At very low temperatures \( T \rightarrow 0 \) K, however, the Landauer principle is not valid because of quantum entanglement [8]. Recently measured energetic demand for single bit processing in conventional computers (Pentium II, 400 MHz) amounts to \( \sim 8.5 \times 10^{-11} \) J bit\(^{-1} \) [9]. The combination of Equations (1.1) and (1.5) yields a mass equivalent of information amounting to \( \sim 3 \times 10^{-38} \) kg bit\(^{-1} \).

The above definition of information assumes a finite number of distinct events (e.g. transmission of characters) and so may somehow represent a digitalized form of information. The digital representation of information, along with Boolean logic (vide infra) constitutes the theoretical basis for all contemporary computing systems, excluding the quantum approach.

The strict mathematical definition of information concerns only information in the sense of a stream of characters and other signals, and is not related to its meaning. There are, however three distinct levels at which information may have different meanings. According to Charles S. Pierce and Charles W. Morris we can define three levels of information: syntactic level, semantic level and pragmatic level. Information on the syntactic level is concerned with the formal relation between the elements of information, the rules of corresponding language, the capacity of communication channels and the design of coding systems for information transmission, processing and storage. The meaning of information and its practical meaning are neglected at this level. The semantic level relates information to its meaning, and semantic units (words and groups of words) are assigned more or less precisely to their meaning. For correct information processing at the syntactic level semantics are not necessary. On the pragmatic level the information is related to its practical value. It strongly depends on the context and may be of economical, political or psychological importance. Furthermore, at the pragmatic level the information value is time dependent and its practical value decreases with time, while correct prediction of future information may be of high value [10,11].

### 1.3 Principles of Boolean Algebra

One of the definitions of the amount of information (Equation 1.2) in the case of \( r = 2 \) implies that the total information contained in a system or event can be expressed using two symbols, for example binary digits. This situation is related to prepositional calculus,
where any sentence has an attributed logic value: TRUE or FALSE. Therefore Boolean algebra based on a two-element set and simple operators can be used for any information processing. Unlike algebra, Boolean algebra does not deal with real numbers, but with the notions of truth and falsehood. These notions, however, are usually assigned symbols of 0 and 1, respectively. This is the most common symbolic representation, but others (e.g. ⊥, ⊤; TRUE, FALSE) are also in use. Furthermore, the numerical operations of multiplication, addition and negation are replaced with the logic operations of conjunction (\(\land\), AND, logic product), disjunction (\(\lor\), OR, logic sum) and complement (\(\neg\), NOT). Interestingly, the same structure would have algebra of the integers modulo 2; these two algebras are fully equivalent \[12,13\]. The operation can be easily defined if Boolean algebra is understood as the algebra of sets, where 0 represents an empty set and 1 a complete set. Then, conjunction is equivalent to the intersection of sets and disjunction to the union of sets, while complement is equivalent to the complement of a set \[10\]. These operations can be simply illustrated using Venn diagrams (Figure 1.2).

Conjunction in Boolean algebra has exactly the same properties as multiplication in algebra. If any of the arguments of the operation is 0 (i.e. FALSE) the operation yields 0, while if both arguments are equal to 1, the result of conjunction is also 1. Disjunction, unlike addition, yields 1 if both arguments are unity, while in other cases its properties are similar to addition. The properties of the complement operation can be described as follows \(1.8\), \(1.9\):

\[
x \land \neg x = 0 \quad (1.8)
\]

\[
x \lor \neg x = 1 \quad (1.9)
\]

Put simply, this operation exchanges the two Boolean values, that is \(\neg 0 = 1\) and \(\neg 1 = 0\). Therefore the double complement yields the initial logic value \(1.10\):

\[
\neg\neg x = x \quad (1.10)
\]

**Figure 1.2** Venn diagrams of set A (a), its complement (b), and the union (disjunction) (c) and intersection (conjunction) (d) of two sets, A and B.
Boolean algebra is based on a set of axioms: associativity, commutativity, distributivity, absorption and idempotence. Furthermore, it assumes the existence of neutral elements and annihilator elements for binary operators.

The associativity rule states that the grouping of the variables in disjunction and conjunction operations does not change the result (1.11), (1.12):

\[(a \lor b) \lor c = a \lor (b \lor c)\]  
\[(a \land b) \land c = a \land (b \land c)\]

Moreover, the operations of disjunction and conjunction are commutative, that is the result does not depend on the order of arguments (1.13), (1.14):

\[a \lor b = b \lor a\]  
\[a \land b = b \land a\]

Both operations are distributive over the other one, that is (1.15), (1.16):

\[a \land (b \lor c) = (a \land b) \lor (a \land c)\]  
\[a \lor (b \land c) = (a \lor b) \land (a \lor c)\]

While the first distributivity law (1.15) is rather intuitive and true also in ordinary algebra, the other (1.16) is true only in Boolean algebra.

The absorption law is an identity linking a pair of binary operations (1.17):

\[a \land (a \lor b) = a \lor (a \land b) = a\]

Therefore Boolean algebra with two elements (0 and 1) and two commutative and associative operators (\(\lor\) and \(\land\)), which are connected by the absorption law, is a lattice. In every lattice the following relation is always fulfilled (1.18):

\[a \lor b = b \iff a \land b = a\]

Therefore the ordering relation ‘\(\leq\)’ can be defined as follows (1.19):

\[a \leq b \iff a \lor b = b\]

The \(\lor\) and \(\land\) operators can be defined as infimum and supremum of sets of arguments, respectively (1.20), (1.21):

\[a \land b \equiv \inf(a, b)\]  
\[a \lor b \equiv \sup(a, b)\]

While in binary logic this is quite intuitive, this analysis is necessary to understand basic binary operators in ternary and higher logic systems (vide infra). The binary Boolean
operators are *idempotent*, that is when applied (many times) to one logic variable, its logic value is preserved (1.22), (1.23):

\[
a \land a = a \\
a \lor a = a
\]  

(1.22)  

(1.23)

For each binary operator there exists a neutral element, which does not change the value of the logic variable. For disjunction this element is 0, while for conjunction it is 1 (1.24), (1.25):

\[
a \lor 0 = a \\
a \land 1 = a
\]  

(1.24)  

(1.25)

**Annihilators** are the elements that destroy information contained in Boolean variables (1.26), (1.27):

\[
a \land 0 = 0 \\
a \lor 1 = 1
\]  

(1.26)  

(1.27)

Boolean algebra is dual, the exchange of 0 and 1 or $\land$ and $\lor$ operators will also result in also a Boolean algebra, but with different properties. Concomitant interchange of values and operators, however, yields the same algebra. This so-called *De Morgan duality* can be formulated as follows (1.28), (1.29):

\[
(\neg a) \lor (\neg b) = \neg(a \land b) \\
(\neg a) \land (\neg b) = \neg(a \lor b)
\]  

(1.28)  

(1.29)

De Morgan duality has very important practical consequences. It allows construction of any Boolean function from only two operators: $\neg$, and either $\lor$ or $\land$. This is especially important in electronics, because the realization of any combinatorial function requires only a limited number of building blocks (e.g. all binary functions can be achieved via combination of several NAND gates, *vide infra*).

### 1.4 Digital Information Processing and Logic Gates

#### 1.4.1 Simple Logic Gates

The simple rules discussed in preceding sections allow any binary logic operations to be performed, and all the complex logic functions can be produced using the basic set of functions: *OR*, *AND* and *NOT*. It is not important whether the information is encoded as electric signals (classical electronics), light pulses (*photronics*) or mechanical movements. The only important issues are the distinguishability of signals assigned to logic values,
and the principles of Boolean algebra. Any physical system whose state can be described as a Boolean function of input signals (also Boolean in nature) is a logic gate. Therefore it is not important if the signals are of electrical, mechanical, optical or chemical nature [14]. Information can be represented by transport of electric charge (classical electronics), ionic charge (electrochemical devices), mass, electromagnetic energy and so on. Furthermore, the actual state of any physical system can be also regarded as a representation of information, for example electrical charge, spin orientation, magnetic flux quantum, phase of an electromagnetic wave, chemical structure or mechanical geometry [15]. Usually the term ‘logic gate’, however, is associated with electronic devices capable of performing Boolean operations on binary variables.

There are two types of one-input electronic logic gates: YES (also called a buffer) and NOT. The YES gate transfers the unchanged signal from the input to the output, while the NOT gate computes its complement (Table 1.1).

There are 16 possible combinations of binary two-input logic gates, but only eight of them have any practical meaning. These include OR, AND and XOR, as well as their combinations with NOT: NOR, NAND, XNOR, INH and IMP (Table 1.2).

The OR gate is one of the basic gates from which all other functions can be constructed. The OR gate produces high output when any of the inputs is in the high state and the output is low when all the inputs are in the low state. Therefore the gate detects any high state at any of the inputs. It computes the logic sum of input variables, that is it performs the disjunction operation.

The AND gate is another of the principal logic gates, it has two or more inputs and one output. The AND gate produces high output (logical 1) only when all the inputs are in the high state. If any of the inputs is in the low state the output is also low (Figure 1.6b). The main role of the gate is to determine if the input signals are simultaneously true. Other words, it performs the conjunction operation or computes the logic product of input variables.

A more complex logic function is performed by exclusive-OR (XOR) gate. This is not a fundamental gate, but it is actually formed by a combination of the gates described above (usually four NAND gates). However, due to its fundamental importance in numerous applications, this gate is treated as a basic logic element and it has been assigned a unique

<table>
<thead>
<tr>
<th>Table 1.1</th>
<th>Truth tables, symbols and Venn diagrams for YES and NOT binary logic gates.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Input</td>
</tr>
<tr>
<td>YES</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NOT</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
The XOR gate yields a high output when the two input values are different, but yields a low output when the input signals are identical. The main application of the XOR gate is in a binary half-adder, a simple electronic circuit enabling transition from Boolean logic to arithmetic.

<table>
<thead>
<tr>
<th>Name</th>
<th>Input A</th>
<th>Input B</th>
<th>Output</th>
<th>Symbol</th>
<th>Venn diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>⊕</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td>AND</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>⋅</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>⋅</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>⋅</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>⋅</td>
<td></td>
</tr>
<tr>
<td>XOR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>⊕</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td>NOR</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>⏹</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>⏹</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>⏹</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>⏹</td>
<td></td>
</tr>
<tr>
<td>NAND</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>⏹</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>⏹</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>⏹</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>⏹</td>
<td></td>
</tr>
<tr>
<td>XNOR</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>⊕</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td>INH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>⋅</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>⋅</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>⋅</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>⋅</td>
<td></td>
</tr>
<tr>
<td>IMP</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>?</td>
<td>A ⨿ B</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
The whole family of logic gates is formed by the concatenation of OR, AND and XOR gates with the NOT gate, which can be connected to the input or output of any of the above gates. The various connection modes and resulting gates are presented in Table 1.2. Gates resulting from concatenation with NOT are obviously not basic gates, but due to their importance they are usually treated as fundamental logic gates together with NOT, OR and AND logic gates.

Along with fundamental and NOT-concatenated devices (Table 1.2) there are several other devices which are not fundamental (or cannot even be described in terms of Boolean logic), but are important for construction of both electronic and non-classical logic devices.

The FAN-OUT operation drives a signal transmitted through one line onto several lines, thus directing the same information into several outputs (Figure 1.3a). A SWAP gate (Figure 1.3b) is a two-input two-output device; it interchanges values transmitted through two parallel lines. This device is especially interesting from the point of view of reversible computation, as it is an element of the Fredkin gate (vide infra). While the FAN-OUT and SWAP operations are seen to be extremely simple devices in electronic implementations (forked connectors and crossed insulated connectors, respectively), in molecular systems it is not that straightforward. FAN-OUT, for example, requires replication of the signalling molecule [16]. A useful device, which is regarded as universal in quantum computing with cellular automata, is the MAJORITY gate (Figure 1.3c). This is a multiple-input single output device. It performs the MAJORITY operation on input bits, that is yields 1 if more than 50% of inputs are in the high state, otherwise the output is zero. A three-input majority gate can be regarded as a universal gate as it can be easily transformed into OR and AND gates (vide infra).

1.4.2 Concatenated Logic Circuits

Single logic gates, even with multiple inputs, allow only basic logic operations on single bits of information. More complex operations, or on larger sets of bits require more complex logic systems. These systems, usually called combinatorial circuits, are the result of connecting several gates. The gates must, however, be connected in a way that eliminates all possible feedback loops, as the state of the circuit should depend only on the input data, not on the device’s history. The most important circuits are the binary half-adder and half-subtractor, and the full adder (Figure 1.4a) [17]. These circuits enable arithmetic operations on bits of information in a binary fashion, which is one of the pillars on which all information technology has been built.

The half-adder is a device composed of two gates: AND and XOR. It has two inputs (two bits to be added) and two outputs (sum and carry). The half-subtractor is a related circuit (the only difference lies in one NOT gate at input) which performs the reverse

![Figure 1.3 Schematics of FAN-OUT (a), SWAP (b) and MAJORITY (c) gates.](image-url)
operation: it subtracts the value of one bit from the other yielding one bit of difference and one bit of borrow (Figure 1.4b).

An interesting device, closely related to the half-adder and the half-subtractor is the binary comparator. It takes two bit inputs (x and y) and yields two bit outputs, which are determined by the relationship between the input quantities. If \( x = y \) one output is set to high (identity bit) and the other to low (majority bit). If \( x > y \) the identity bit is zero, while the majority bit equals 1. In the case of \( x < y \) both output bits are 0 (Table 1.3, Figure 1.4c).

The appropriate connection of two binary half-adders or binary half-subtractors results in two more complex circuits, the binary adder and the binary subtractor, respectively. A full adder consists of two half-adders and an OR gate (Figure 1.5a). The circuit performs full addition of three bits yielding two-bit results. Similarly, a full subtractor is built from two half-subtractors and an OR gate (Figure 1.5b). This device can subtract three one-bit numbers yielding a two-bit binary result. The schematics of the binary full adder and full subtractor are shown in Figure 1.5 and the corresponding logic values in Table 1.4.

The simple concatenated logic circuits show the infinite possibilities of combinations of simple building blocks (logic gates) into large functional circuits.

### 1.4.3 Sequential Logic Circuits

A circuit comprised of connected logic gates, devoid of feedback loops (memory), is a combinatorial logic circuit, a device whose output signal is a unique Boolean function of input variables. A combinatorial logic circuit with added memory forms a sequential logic circuit.

<p>| Table 1.3 Truth table for binary half-adder, half-subtractor and comparator. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
| Input | Half-adder | Half-subtractor | Comparator |</p>
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>(c,s)&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Decimal value</th>
<th>(b,d)&lt;sup&gt;b&lt;/sup&gt;</th>
<th>Decimal value</th>
<th>(i,m)&lt;sup&gt;c&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(0,0)</td>
<td>0</td>
<td>(0,0)</td>
<td>0</td>
<td>(1,0)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>(0,1)</td>
<td>1</td>
<td>(1,1)</td>
<td>-1</td>
<td>(0,0)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>(0,1)</td>
<td>1</td>
<td>(0,1)</td>
<td>1</td>
<td>(0,1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(1,0)</td>
<td>2</td>
<td>(0,0)</td>
<td>0</td>
<td>(1,0)</td>
</tr>
</tbody>
</table>

<sup>a</sup>carry, sum.<br>
<sup>b</sup>borrow, difference.<br>
<sup>c</sup>identity, majority.
circuit, often referred to as an automaton (Figure 1.6). Memory function can be simply obtained by the formation of a feedback loop between the outputs and inputs of individual gates within the circuit. The output state of an automaton depends on the input variables and the inner state (memory) of the device.

The memory function can be simply realized as a feedback loop connecting one of the outputs of the logic device to one of the inputs of the same device. The simplest (but not very useful) memory cell can be made on the basis of an OR gate via feeding back the output to the input (Figure 1.7).

Figure 1.5  Electronic diagrams for binary full adder (a) and full subtractor (b). HA stands for half-adder and HS for half-subtractor, respectively. In the case of subtractors, a stands for subtrahend and b for minuend.

Figure 1.6  Schematic of a sequential information processing device (an automaton). The simplest memory for the device can be realized by a feedback loop (dashed arrow), feeding some of the outputs of the device to the input.
Initially the device yields an output of 0. However, when the input is set to high, the output also switches to the high state. As the output is directed back to the input, the device will remember the state until power-off. Loops involving XOR and NAND gates tend to generate oscillations. These oscillations render the circuits unusable. This problem, however, can be simply solved in feedback circuits consisting in two gates (NOR, NAND, etc.) and two feedback loops (Figure 1.7b). This circuit is the simplest example of a latch (flip-flop), a device, the state of which is a Boolean function of both the input data and the state of switch. This device can serve as a simple memory cell and after some modification can be used as a component of more complex circuits: shift registers, counters, and so on.

The two inputs of the latch, named R and S (after set and reset) change the state of the outputs in a complex way (Table 1.5), provided they are never equal to 1 at the same time (i.e. R = S = 1) as this particular combination of inputs results in oscillations of the latch.

In the case of most input combinations, the output state of the device is not changed, but the (1,0) state induces 0 → 1 switching, while the (0,1) state results in 1 → 0 switching.

<table>
<thead>
<tr>
<th>Input</th>
<th>Adder</th>
<th>Subtractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c/p&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<sup>a</sup>carry/pay back.  
<sup>b</sup>(carry, sum).  
<sup>c</sup>(borrow, difference).

*Figure 1.7*  Looped OR gate as a model of the simplest memory cell (a) and RS-type flip-flop built from two NOR gates (b).
1.5 Ternary and Higher Logic Calculi

Binary logic can be generalized for any system with a finite number of orthogonal logic states. In ternary logic any sentence may have three different values: FALSE, TRUE or UNKNOWN. Analogous to binary logic, numerical values can be associated with these values, as shown in Table 1.6.

Logic operations are defined in an analogous way to the case of binary logic:

The unary ternary operator NOT is defined as (1.30):

$$\neg a \equiv 1 - a$$  \hspace{1cm} (1.30)

while the binary ternary operators are defined as follows:

$$a \land b \equiv \inf(a, b)$$  \hspace{1cm} (1.31)

$$a \lor b \equiv \sup(a, b)$$  \hspace{1cm} (1.32)

$$a \otimes b \equiv \sup[\inf(a, 1 - b), \inf(1 - a, b)]$$  \hspace{1cm} (1.33)

The logic values for unary and binary ternary operators are shown in Tables 1.7 and 1.8.

In any multivalued logic the unary and binary operators can be defined as follows:

$$\neg a \equiv T_0 - a$$  \hspace{1cm} (1.34)

$$a \land b \equiv \inf(a, b)$$  \hspace{1cm} (1.35)

### Table 1.6 Numerical representations of ternary logic values in unbalanced and balanced system.

<table>
<thead>
<tr>
<th>Logic value</th>
<th>Numerical representation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unbalanced</td>
</tr>
<tr>
<td>FALSE</td>
<td>0</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>$\frac{1}{2}$, #</td>
</tr>
<tr>
<td>TRUE</td>
<td>1</td>
</tr>
</tbody>
</table>