# Scalable Thue Morse (TM) Logic/Algebra, Part 1

Peter Hettich, Alexis De Vos

June 4, 2025

**Keywords**: computer science, Thue Morse (TM) Sequence, scalable (TM) Boolean Algebra, scalable (TM) Huntington System, scalable (TM) Switching algebra, formal metasystems, modern computer architectures (MCA)

**MSC**: 68M07; 11B85; 06E30; 06E75

**Abstract**

All logical/algebraic systems relevant to computer science, from Aristotle to C. E. Shannon, are scaled into metasystems, new multidimensional, higher-value, complementary (TM) logics/algebras. Claude E. Shannon's Boolean Switching Algebra and its application to the design of digital computers is a fundamental, physical proof of Boolean logic. Accordingly, the scalable Thue-Morse (TM) logic/algebra is transformed into extended technical structures to realize combinatorial (Part 2) and sequential circuit arrangements (Part 3). This achieves a practical validation of higher-order Thue-Morse (TM) logic/algebra. This enables the reduction and computation of complex logical many particle systems and the application in future modern computer architectures (MCA).

## 1 Introduction

All logic systems, from Aristotle, G. Boole, B. Russel (Theory of Types), Y. Huntington (Huntington Axiom System), K. Gödel (Incompleteness Theorems), to sequential logic (various authors), are based on the assignment of single truth values – either false (f) or 0, or true (t) or 1 – to statement variables. A fundamental principle of these systems is the exclusivity of truth value assignment, according to which a formula, a correctly constructed and meaningful string of characters, cannot simultaneously assume both values (f and t, or 0 and 1); the coexistence of (ft), (tf), or (01), (10), the contradiction, as valid value assignments is excluded in these frameworks.

Classical and modern Computer Architectures (MCA) fundamentally operate on the basis of one-dimensional Boolean logic or algebra and binary states (0,1), which are physically represented by the state of a single line. The representation by single binary variables limits the basic expressiveness to the representation of point-like entities in the state space.

Complex systems constitute themselves as many-particle systems whose components are organized in network-like architectures. A defining characteristic of these systems is their ability to coherent state changes. The expressiveness of Boolean logic or algebra proves inadequate for the formal analysis of complex, higher-dimensional, and multi-valued systems.

From the limitation to binary, one-dimensional logic values result two fundamental challenges for digital technology and computer science: Firstly, there is a lack of established formal algebraic hardware structures for the comprehensive treatment of higher-dimensional, multi-valued combinatorial logic. Formal metasystems do not yet exist. Secondly, the formal reduction of sequential circuit arrangements to purely algebraic frameworks is not possible. Metasystems or sequential circuit arrangements cannot yet be reduced to Boolean logic/algebra.

The fundamental principles of established logics are therefore recursively scaled and extended to a new, multidimensional, higher-valued, complementary TM logic.

The scaled positive TM logic offers a theoretical foundation for the analysis and synthesis of complex systems. It recursively scales and extends classical logical formalisms into higher-dimensional, multi-valued, and complementary structures, overcoming their limitations. Recursive scaling is the proof method of the new logic. These extended structures provide design rules for combinatorial (Part 2) and sequential circuit arrangements (Part 3), which form the basis for the construction of TM computers and complex KTM computers. This results in higher-dimensional switching networks that aggregate multiple elementary units (n cubes) into cohesive entities. With these higher-dimensional n-cube networks, one can calculate and reason logically. This makes two fundamental challenges of digital technology and computer science solvable: Firstly, the higher-dimensional combinatorial switch arrangements enable decidable, complete, provable metasystems that formally capture deeper structural properties. Secondly, the higher-dimensional sequential switch arrangements allow a consistent reduction of flip-flops and algorithms to algebraic structures of Boolean algebra.

The new formal TM logic is based on the analysis of logic systems [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14], mathematics [15] [16] [17], natural science [18] [19] [20] [21], digital technology [22] [23] [24] [25] [26], known scaling principles from the development of Artificial Intelligence [27] [28] and quantum computers [29] [30], from the analysis of the Thue-Morse sequence, and the analysis of recursive formulas.

## 2 Scalable Thue Morse (TM) Logic/Algebra

The essential principles of the scalable Thue Morse (TM) Logic/Algebra can be constructed from the (TM) sequence.

The (TM) sequence is encountered as a characteristic feature of complex problems in all complex areas of science, technology, and society. This area includes, for example, classical physics [31] [32] [33], quantum physics [34] [35] [36] [37], DNA analysis [38] [39] [40] [41], mathematics [42] [43] [44] [45], economics [46], art [47] [48], linguistics [49] [50] [51], or generally complex systems [52] [53] [54]. Topologically, the TM sequence can be mapped onto n-cubes [55]. A simple string of characters shows enormous diversity and complexity. The (TM) sequence as a meta structure becomes visible in almost all complex structures of human, rational perception.

Technically, the Thue-Morse sequence is a common structure, a code, or a common feature that connects all modern computer architectures (MCA) from quantum physics to human DNA. The research on Dual Rail Reversible Computing (DRRC) and Quantum Computing (QC) [56] [57] represents a crucial link to the new scalable Thue Morse (TM) Logic/Algebra.

For classical and modern Computer Architectures (MCA) [58], such as Dual Rail Computing (DRC), Dual Rail Reversible Computing (DRRC), Quantum Computing (QC), DNA Computing (DNAC), Biocomputing (BC), Neural Networks (NN), and Cryptography (KR), higher logic/algebra systems do not yet exist.

The following explanation uses the terms AND, OR, NOT and the apostrophe A' for the logical operations to indicate the negation: NOT A. This notation is equivalent to the notation with the symbols (∧, ∨, ¬).

The (TM) sequence forms the basis for scaling a basic system of Boolean logic/algebra (*BAS),*

*BAS 1*: (0,1, AND, OR, NOT). Any formula such as P = A AND B of a Boolean algebra in positive logic $P_L$ has a dual, complementary formula P' = A' OR B' in negative logic $N_L$ which is obtained by replacing 0 with 1 and AND with OR and vice versa. Both formulas can be combined into a new formula: PP' = AA' AND BB'. The new value tables are created by replacing 0 with 01 and 1 with 10. The variables P and A are created by supplementing with the complement such as PP' and AA'. The fundamental law is a recursive formula with which higher and higher BITs: (0,1), (01,10), (0110,1001) ... and higher variables: A, AA', AA'A'A... functions can be formed.

New basic systems *BAS 2*: (01,10, AND, OR, NOT), *BAS 3*: (0110,1001, AND, OR, NOT), a higher-dimensional, multi-valued, complementary Boolean logic/algebra are the result. These new Boolean structures can be connected to an overall system using XOR.

**3 (TM) Sequence**

In mathematics, the (TM) sequence or Prouhet Thue Morse sequence is the binary sequence (an infinite sequence of zeros and ones) obtained by starting at 0 and successively appending the Boolean complement to the sequence obtained so far. The first steps of this procedure result in the strings 0, then 01, 0110, 01101001, 0110100110010110, etc ...

The sequence is named after Axel Thue and Marston Morse.

**4 (TM) Elements**

The elements $E_n$: $E_0$, $E_1$, $E_2$, $E_3$… of the (TM) Sequence.

$E_0 = 0$,

$E_1 = 01$,

$E_2 = 0110$,

$E_3 = 01101001$…

They are generated recursively by Boolean complement formation. The recursive definition is called scaling.

The scaling principle of the elements:

$E_{n+1} = E_n E_n‘$.

For n = 0 and $E_0 = 0$ results in the complement to $E_0‘ = 1$, it follows from this $E_1 = E_0E_0‘ = (01)$.

The string (01) is a concatenation, summary of the elements $E_n = 0$ with the complement $0‘ = E_n‘ = 1$ to a whole (01).

**5 (TM) BITs**

The elements of the string that are greater than 0 are BITs with a lower value $K_n$ and an upper complementary value $K_n‘$.

The scaling principle for the (TM) BITs:

$K_{n+1} = K_n,K_n‘$

with $K_n‘$ as a Boolean complement to $K_n$.

The (TM) BITs yield:

$K_1 = (0,1)$,

$K_2 = (01,10)$,

$K_3 = (0110, 1001)$,

$K_4 = (01101001,10010110)$ …

whereby the higher values can be considered higher-value BITs or higher elementary decisions.

The (TM) BITs in decimal representation results in higher, decimal values or multi-valued BITs:

$K_1 = (0,1)$,

$K_2 = (1,2)$,

$K_3 = (6,9)$,

$K_4 = (105,150)$ …

The (TM) BITs can be mapped to dimensions $D_{n.}$

$D_1$: (0,1),

$D_2$: (01,10),

$D_3$:(0110, 1001),

$D_4$:(01101001, 10010110)…

**6 (TM) Variables**

The scaling principle for variables:

$V_{n+1} = V_n (V_n)$‘ with $(V_n)'$ as a Boolean complement to $V_n$. For n = 0, defined $V_1 = A$.

The scaling sequence:

$V_1 = A$,

$V_2 = AA$‘,

$V_3 = AA'A'A$,

$V_4 = AA'A'AA'AAA'$…

The (TM) variables are assigned to dimensions D.

$D_1$: A,

$D_2$: AA‘,

$D_3$: AA’A’A,

$D_4$: AA’A’AA’AAA‘…

The variables represent the (TM) BITs.

A → (0,1),

AA‘ → (01,10),

AA’A‘A → (0110,1001),

AA’A’AA’AAA‘ → (01101001,10010110).

The assignment in decimal representation:

A → (0,1),

AA‘→ (1,2),

AA’A’A → (6,9),

AA’A’AA’AAA‘→ (105,150).

**7 (TM) Functions**

The scaling principle for functions:

$f_{n+1} = f_n (f_n)$‘ with $(f_n)$‘ as a Boolean complement to $f_n$. For n = 0, defined $f_1 = f$ .

The (TM) Function sequence results in:

$f_1$ = f,

$f_2$ = f (f)‘,

$f_3$ = f (f)‘ (f)‘f,

$f_n$ = f($f_n$)‘ ($f_n$)‘f ($f_n$)‘ff($f_n$)‘…

The "IDENT" function for a variable is uniquely determined by the operator IDENT: P = (A):

$IDENT_{n+1} = IDENT_n (IDENT_n)$‘:

The scaling sequence:

$f_1$ = P = A,

$f_2$ = PP‘ = AA‘,

$f_3$ = PP’P’P = AA’A’A,

$f_4$ = PP’P’PP’PPP‘ = AA’A’AA’AAA‘…

Recursion can be extended to functions with multiple variables, as shown by the example of the AND function f: P = A AND B. The scaling sequence requires one operator.

Scaling principle for the AND function.

$AND_{n+1} = AND_n (AND_n)$‘:

The scaling sequence:

$f_1$ = P = A AND B,

$f_2$ = PP‘ = AA‘ AND BB‘,

$f_3$ = PP’P’P = AA’A’A AND BB‘B’B,

$f_n$ = PP’P’PP’PPP‘… = AA’A’AA’AAA‘… AND BB‘B’BB’BBB‘…

Scaling principle for the XOR function:

$XOR_{n+1} = XOR_n (XOR_n)$‘

The scaling sequence:

$f_1$ = P= A XOR B,

$f_2$ = PP‘ = AA‘ XOR BB‘,

$f_3$ = PP’P’P = AA’A’A XOR BB‘B’B,

$f_n$ = PP’P’PP’PPP‘… = AA’A’AA’AAA‘… XOR BB‘B’BB’BBB‘…

Functions can also be represented in a table of values. The tabular form is the easiest way to represent the scaling of Boolean algebra to higher dimensions. In the table on the left, enter the input variables (A, B, C...) and on the right, enter the output variables (P, Q, R...). If you start with the single-digit constants (0,1) for a BIT of dimension 1 in the classical Boolean function, the columns are single-digit and contain the values 0 or 1 in an ordered form. In the next step, the columns are doubled and the complementary or dual function is added to each function, i.e. a "0" becomes "01", a 1 becomes "10", the variables are dealt with accordingly: A becomes AA', P becomes PP'. This results in a Boolean algebra of dimension 2. In the next step, dimension 3, 01 becomes 0110 and 10 becomes 1001, AA' becomes AA'A'A and PP' becomes PP'P'P. The BIT values and the variables in the value table double in each dimension.

## 8 (TM) System

The functions $f_n$ of the individual dimensions $D_n$ of a system S can be combined by XOR to form a new hyperdimensional overall system $S_n$: $D_1$ XOR $D_2$ XOR $D_3$ XOR... $D_n$, Part 3.

## 9 (TM) Boolean Axioms and rules

The most important rules and axioms of the Boolean algebra of dimension 1 can be scaled to higher Dimensions Dn.

D1: $BAS_1$ = [0,1, AND, OR, NOT].

Commutative Laws:

A AND B = B AND A for all A, B ∈ *BAS.*

A OR B = B OR A for all A, B ∈ *BAS.*

Associative laws:

(A AND B) AND C = A AND (B AND C) for all A, B, C ∈ *BAS.*

(A OR B) OR C = A OR (B OR C) for all A, B, C ∈ *BAS.*

Distributivity laws:

A AND (B OR C) = (A AND B) OR (A AND C) for all A, B, C ∈ *BAS.*

A OR (B AND C) = (A OR B) AND (A OR C) for all A, B, C ∈ *BAS.*

Absorption laws:

A AND (A OR B) = A for all A, B ∈ *BAS*.

A OR (A AND B) = A for all A, B ∈ *BAS*.

Existence of negation:

There exists an element A' ∈ B for every A ∈ s *BAS* such that:

A AND A'= 0 for all A ∈ *BAS*.

Existence of neutral elements:

There are two elements 0 and 1 ∈ *BAS*, so that:

A AND 1 = A for all A ∈ *BAS*.

A OR 0 = A for all A ∈ *BAS*.

Defining axiom of double negation:

A'' = A for all A ∈ *BAS*.

Statement of contradiction:

A AND A' = 0.

Statement of completeness:

A OR A' = 1.

Extension of the axiom system:

This axiom system can be extended by other axioms to describe additional properties of Boolean algebra, e.g. the idempotent laws, the laws of De Morgan.

Example:

Idempotent law for the conjunction:

A AND A = A for all A ∈ *BAS*.

De Morgan's law for the conjunction:

(A AND B)' = A' OR B' for all A, B ∈ *BAS*.

D2: $BAS_2$ = [01,10, AND, OR, NOT].

Commutative laws:

AA' AND BB' = BB' AND AA' for all AA', BB' ∈ *BAS*.

AA' OR BB' = BB' OR AA' for all AA', BB' ∈ *BAS*.

Associative laws:

(AA' AND BB') AND CC' = AA' AND (BB' AND CC') for all AA', BB', CC'∈ *BAS*.

(AA' OR BB') OR CC' = AA' OR (BB' OR CC') for all AA', BB', CC' ∈ *BAS*.

Distributivity laws:

AA’ AND (BB’ OR CC’) = (AA’ AND BB’) OR (AA’ AND CC’) for all AA’, BB’, CC’ ∈ *BAS*.

AA’ OR (BB’ AND CC’) = (AA’ OR BB’) AND (AA’ OR CC’) for all AA’, BB’, CC’ ∈ *BAS*.

Absorption laws:

AA’ AND (AA’ OR BB’) = AA’ for all AA’, BB’ ∈ *BAS*.

AA’ OR (AA’ AND BB’) = AA’ for all AA’, BB’ ∈ *BAS*.

Existence of negation:

There exists an element A'A ∈ *BAS* for each AA' ∈ *BAS*, such that:

AA’ AND A’A = 01 for all AA’ ∈ *BAS*.

Existence of neutral elements:

There are two elements 01 and 10 ∈ *BAS*, so that:

AA’ AND 10 = AA’ for all AA’ ∈ *BAS*.

AA’ OR 01 = AA’ for all AA’ ∈ *BAS*.

Defining axiom of double negation:

(AA’)’’ = AA’ for all AA’ ∈ *BAS*.

Statement of contradiction:

AA’ AND A’A = 01.

Statement of completeness:

AA’ OR A’A = 10.

Extension of the axiom system:

This axiom system can be extended by other axioms to describe additional properties of Boolean algebra, e.g. the idempotent laws, the laws of De Morgan.

Example:

Idempotent law for the conjunction:

AA’ AND AA’ = AA’ for all AA’ ∈ *BAS*.

De Morgan's law for the conjunction:

(AA’ AND BB’)’ = A’A OR B’B for all AA’, BB’ ∈ *BAS*.

D3: $BAS_3$ = [0110,10001, AND, OR, NOT]

Commutative laws:

AA’A’A AND BB’B’B = BB’B’B AND AA’A’A for all AA’A’A, BB’B’B ∈ *BAS*

AA’A’A OR BB’B’B = BB’B’B OR AA’A’A for all AA’A’A, BB’B’B ∈ *BAS.*

Associative laws:

(AA’A’A AND BB’B’B) AND CC’C’C = AA’A’A AND (BB’B’B AND CC’C’C) for all AA’A’A, BB’B’B, CC’C’C ∈ *BAS.*

(AA’A’A OR BB’B’B) OR CC’C’C = AA’A’A OR (BB’B’B OR CC’C’C) for all AA’A’A, BB’B’B, CC’C’C ∈ *BAS.*

Distributivity law

AA’A’A AND (BB’B’B OR CC’C’C) = (AA’A’A AND BB’B’B) OR (AA’A’A AND CC’C’C) for all AA’A’A, BB’B’B, CC’C’C ∈ *BAS.*

AA’A’A OR (BB’B’B AND CC’C’C) = (AA’A’A OR BB’B’B) AND (AA’A’A OR CC’C’C) for all AA’A’A, BB’B’B, CC’C’C ∈ *BAS.*

Absorption laws:

AA’A’A AND (AA’A’A OR BB’B’B) = AA’A’A for all AA’A’A, BB’B’B ∈ *BAS.*

AA’A’A OR (AA’A’A AND BB’B’B) = AA’A’A for all AA’A’A, BB’B’B ∈ *BAS.*

Existence of negation:

There is an element A’AAA’ ∈ *BAS* for each AA’A’A ∈ *BAS*, so that:

AA’A’A AND A’AAA’ = 0110 for all AA’A’A∈ *BAS.*

Existence of neutral elements:

There are two elements 0110 and 1001 ∈ *BAS*, so that.

AA’A’A AND 1001 = AA’A’A for all AA’A’A ∈ *BAS,*

AA’A’A OR 0110 = AA’A’A for all AA’A’A ∈ *BAS.*

Defining axiom of double negation:

(AA’A’A)’’ = AA’A’A for all AA’A’A ∈ *BAS.*

Statement of contradiction:

AA’A’A AND A’AAA’ = 0110.

Statement of completeness:

AA’A’A OR A’AAA’ = 1001.

Extension of the axiom system:

This axiom system can be extended by other axioms to describe additional properties of Boolean algebra, e.g. the idempotent laws, the laws of De Morgan.

Example:

Idempotent law for the conjunction:

AA’A’A AND AA’A’A = AA’A’A for all AA’A’A ∈ *BAS.*

De Morgan's law for the conjunction:

(AA’A’A AND BB’B’B)’ = A’AAA’ OR B’BBB’ for all AA’A’A, BB’B’B ∈ *BAS.*

D4: $BAS_4$ = [01101001, 10010110, AND, OR, NOT].

Commutative laws:

AA’A’AA’AAA’ AND BB’B’BB’BBB’ = BB’B’BB’BBB’ AND AA’A’AA’AAA’ for all AA’A’AA’AAA’, BB’B’BB’BBB’ ∈ *BAS.*

AA’A’AA’AAA’ OR BB’B’BB’BBB’ = BB’B’BB’BBB’ OR AA’A’AA’AAA’ for all AA’A’AA’AAA’, BB’B’BB’BBB’ ∈ *BAS.*

Associative laws:

(AA’A’AA’AAA’ AND BB’B’BB’BBB’) AND CC’C’CC’CCC’ = AA’A’AA’AAA’ AND (BB’B’BB’BBB’ AND CC’C’CC’CCC’) for all AA’A’AA’AAA’, BB’B’BB’BBB’, CC’C’CC’CCC’ ∈ *BAS.*

(AA’A’AA’AAA’ OR BB’B’BB’BBB’) OR CC’C’CC’CCC’ = AA’A’AA’AAA’ OR (BB’B’BB’BBB’ OR CC’C’CC’CCC’) for all AA’A’AA’AAA’, BB’B’BB’BBB’, CC’C’CC’CCC’ ∈ *BAS.*

Distributivity law:

AA’A’AA’AAA’ AND (BB’B’BB’BBB’ OR CC’C’CC’CCC’) = (AA’A’AA’AAA’ AND BB’B’BB’BBB’) OR (AA’A’AA’AAA’ AND CC’C’CC’CCC’) for all AA’A’AA’AAA’, BB’B’BB’BBB’, CC’C’CC’CCC’ ∈ *BAS.*

AA’A’AA’AAA’ OR (BB’B’BB’BBB’ AND CC’C’CC’CCC’) = (AA’A’AA’AAA’ OR BB’B’BB’BBB’) AND (AA’A’AA’AAA’ OR CC’C’CC’CCC’) for all AA’A’AA’AAA’, BB’B’BB’BBB’, CC’C’CC’CCC’ ∈ *BAS.*

Absorption laws:

AA’A’AA’AAA’ AND (AA’A’AA’AAA’ OR BB’B’BB’BBB’) = AA’A’AA’AAA’ for all AA’A’AA’AAA’, BB’B’BB’BBB’ ∈ *BAS.*

AA’A’AA’AAA’ OR (AA’A’AA’AAA’ AND BB’B’BB’BBB’) = AA’A’AA’AAA’ for all AA’A’AA’AAA’, BB’B’BB’BBB’ ∈ *BAS.*

Existence of negation:

There is an element A’AAA’AA’A’A ∈ *BAS* for all AA’A’AA’AAA’ ∈ *BAS*, so that:

AA’A’AA’AAA’ AND A’AAA’AA’A’A = 01101001 for all AA’A’AA’AAA’∈ *BAS.*

Existence of neutral elements:

There are two elements 01101001 and 10010110 ∈, *BAS* so that:

AA’A’AA’AAA’ AND 10010110 = AA’A’AA’AAA’ for all AA’A’AA’AAA’ ∈ *BAS,*

AA’A’AA’AAA’ OR 01101001 = AA’A’AA’AAA’ for all AA’A’AA’AAA’ ∈ *BAS.*

Defining axiom of double negation:

(AA’A’AA’AAA’)’’ = AA’A’AA’AAA’ for all AA’A’AA’AAA’ ∈ *BAS*.

Statement of contradiction:

AA’A’AA’AAA’ AND A’AAA’AA’A’A = 01101001.

Statement of completeness:

AA’A’AA’AAA’ OR A’AAA’AA’A’A = 10010110.

Extension of the axiom system:

This axiom system can be extended by other axioms to describe additional properties of Boolean algebra, e.g. the idempotent laws, the laws of De Morgan.

Example:

Idempotent law for the conjunction:

AA’A’AA’AAA’ AND AA’A’AA’AAA’ = AA’A’AA’AAA’ for all AA’A’AA’AAA’ ∈ *BAS*.

De Morgan's law for the conjunction:

(AA’A’AA’AAA’ AND BB’B’BB’BBB’)’ = A’AAA’AA’A’A OR B’BBB’BB’B’B for all AA’A’AA’AAA’, BB’B’BB’BBB’ ∈ *BAS*.

## 10 Boolean Algebra, Dimension n, $2^N$ Rail

All axioms and rules of Boolean algebra can be extended to any Dn rail system by doubling the D(n-1) system.

The scaling can further be applied to the Huntington axioms to the switching algebra and the basis systems.

## 11 (TM) Huntington System

Huntington's axioms provide a formal definition for Boolean algebra. They define a Boolean algebra: HA = [ K, TEE, RUM, N, O, I] as a set K with two binary operations TEE, RUM (often denoted as '.' for AND and '+' for OR), a unary operation N (complement, often denoted as ' ¬’), and two distinct universal bounds O, I (O for ZERO (0) and I for ONE (1)).

The axioms are fundamental to understanding the structure of Boolean algebra and, by extension, are crucial for the formulation of switching algebra (also known as binary logic), which is a specific two-valued Boolean algebra used to describe the behaviour of digital circuits.

These Huntington's Algebra System (*HAS*) is scaled into higher dimensions Dn:

Dn: $(HAS)_1$ = [K, TEE, RUM, N, O, I].

Example of Boolean algebra:

D1: $(HAS)_1$ = [{0,1}, TEE, RUM, N, 0, 1],

D2: $(HAS)_2$ = [{01,10}, TEE, RUM, N, 01, 10], D3: $(HAS)_3$ = [{0110,1001}, TEE, RUM, N, 0110, 1001],

D4: $(HAS)_4$ = [{01101001, 10010110}, TEE, RUM, N, 01101001, 10010110].

## 12 (TM) Switching algebra

The switching algebra only knows the two values 0 and 1, the two binary operations conjunction (AND) and disjunction (OR) as well as the one binary operation of the negation (NOT). The two neutral elements must be equal to these values because of the binary set of values.

These switching algebra System (*SAS*) is scaled into higher dimensions Dn:

D1: $(SAS)_1$ = [ {0,1}, AND, OR, NOT, 0,1],

D2: $(SAS)_2$ = [ {01,10}, AND, OR, NOT, 01, 10],

D3: $(SAS)_3$ = [ {0110,1001}, AND, OR, NOT, 0110, 1001],

D4: $(SAS)_4$ = [ {01101001,10010110}, AND, OR, NOT, 01101001, 10010110].

Each Basic System can be scaled into a switching algebra.

## 13 (TM) Basic Systems, *BAS*

Basic systems, *BAS* contain the axioms of a special switching algebra.

The normal form theorems and the fundamental theorem of switching algebra establish that the set {AND, OR, NOT} constitutes a complete basis for Boolean functions. This raises the question, motivated by both theoretical and practical considerations, of whether smaller bases, such as binary or even unary ones (like NAND), suffice. The exploration of such bases relies on the axioms and rules governing Boolean algebra.

Basic systems of dimension 1

AND, OR, NOT System,

$BAS_A$ = [0,1, AND, OR, NOT],

Ring Sum System

$BAS_B$ = [0,1, AND, XOR, „1“],

NAND System

$BAS_C$ = [0,1, NAND].

These Basis Systems are scaled into higher dimensions Dn:

AND, OR, NOT System

D1: $BAS_1$ = [0,1, AND, OR, NOT],

D2: $BAS_2$ = [01,10, AND, OR, NOT],

D3: $BAS_3$ = [0110,1001, AND, OR, NOT],

D4: $BAS_4$ = [01101001, 10010110, AND, OR, NOT].

Ring Sum System

D1: $BAS_1$ = [0,1, AND, XOR, „1“],

D2: $BAS_2$ = [01,10, AND, XOR, „10“],

D3: $BAS_3$ = [0110,1001, AND, XOR, „1001“],

D4: $BAS_4$ = [01101001, 10010110, AND, XOR, „10010110“].

NAND System

D1: $BAS_1$ = [0,1, NAND],

D2: $BAS_2$ = [01,10, NAND],

D3: $BAS_3$ = [0110,1001, NAND],

D4: $BAS_4$ = [01101001, 10010110, NAND].

The NAND system is universal. A computer can theoretically be built entirely from combinatorial NAND switching networks and sequential switching functions based on the NAND FLIPFLOP.

All axioms and rules of (TM) Boolean Logic/algebra, (TM) Huntington's axioms, (TM) switching algebra, (TM) basic systems can be extended to 2n rail system.

**14 Conclusion**

Aristotle [2]

The three laws of Aristotle form the fundamental principles of Aristotelian thought and the classical logic systems to date. They lay the foundation for consistent and contradiction-free argumentation and were formulated by him in the Metaphysics.

Aristotle mainly formulated the laws in words. Over time, the formal representation occurred in the form of formulas. These are given here for simplification.

Law of Identity

(principium identitatis)

"It is therefore clear that identity is a kind of unity of being."

Everything is identical with itself: A = A

Law of Non-Contradiction

"Yet the most certain of all basic principles is that regarding which it is impossible to be mistaken. Which this is, we shall now state; for it is impossible for the same thing to belong and not to belong simultaneously to the same thing and in the same respect."

Two mutually contradictory statements cannot be true at the same time: A AND A' = 0

Law of Excluded Middle

(tertium non datur/principium exclusi terti)

"And yet it is not possible that there should be an intermediate between contradictories, but of one or the other it is necessary either to assert or to deny."

A statement is either true or false; a third is excluded: A OR A' = 1

A fundamental principle of logic systems is the exclusivity of truth value assignment, according to which a statement cannot simultaneously assume both truth values (f and w or 0 and 1).

G. Boole [4]

The three laws of Aristotle were adopted in Boolean logic/algebra in the following single-variable form.

The identity function P = A is a manifestation of the law of identity at the level of truth values.

The formula 0 = A AND A' is a manifestation of the law of non-contradiction.

The formula for completeness 1 = A OR A' is a manifestation of the law of excluded middle.

Scaled New TM Logic/Algebra

The scaling of Aristotle's three laws in TM Logic/Algebra clarifies the fundamental difference between classical logic systems and the new TM Logic/Algebra.

Here, the classical, single-variable, positive Boolean logic/algebra (P) of dimension 1 is recursively scaled with the complementary, single-variable, negative Boolean logic/algebra (P' = N) to form a new two-variable, TM positive Boolean logic/algebra (PN) of dimension 2.

The identity PP' = AA' with the two-variable values (ft, tf) or (01, 10) is the new unit. The complementary, single-variable values (f, t) or (0, 1) of dimension 1 are simultaneously represented at two positions. In digital technology, two lines are used for one BIT for this purpose. Complementary values are thereby combined. Contradictory statements A and A' are formally represented simultaneously. This is not possible with classical, single-variable logic.

The three laws then take the following form:

The new identity function PP' = AA' is a manifestation of the law of identity.

The new formula 01 = AA' AND A'A is a manifestation of the law of non-contradiction.

The new formula for completeness 10 = AA' OR A'A is a manifestation of the law of excluded middle.

TM Logic/Algebra can be technically transferred to TM switching algebra by using two C-NOT/XOR gates to build a new AND gate or a NAND gate or a Toffoli gate, Part 2.

The new identity function PP' = AA' is realized by an XOR/C NOT gate. With two XOR/C NOT gates, the AND function PP' = AA' AND BB' can be realized, and a new basic system can be formulated. Dimension 2 (new Dual Rail) is a new, formal metasystem. It is provably complete, consistent, and decidable.

This allows, for example, the undecidability of classical Dual Rail technology to be resolved. Classical Dual Rail technology uses one line and the Single Rail switch. The AND function P = A AND B is realized by positive logic (P) with the constants (0, 1) and negative logic (P' = N) with the constants (1, 0). This can be summarized in a formula P, P' = A, A' AND, OR B, B'. Four values (0, 1) and (1, 0) are used. The function is not decidable and not a Boolean algebra. There is no identity for two-variable values and no binary operators that can be constructed from two identities.

Recursive scaling can be transferred to TM sequential circuits (e.g., NAND Flip-Flop), Part 3.

Scaled, sequential flip-flops can be reduced to a combinatorial, algebraic form. This opens up the possibility of algebraically treating machine language (State Charts) and its algorithms/software. This unifies combinatorial and sequential circuits.

The basis for this is again the new identity function PP' = AA'. With variable constants KK', the XOR function results: PP' = AA' XOR KK' in IN/OUT operation. If feedback is added and PP' is connected to AA', a simple sequential circuit PP'+ = PP' XOR KK' is obtained.

The XOR function can be reversed to OUT/IN operation: KK' = AA' XOR PP'. If feedback is added again and PP' is connected to AA', a combinatorial switch arrangement 01 = PP' XOR PP' is obtained.

This principle is transferred to the Flip-Flop.

With the new scaled combinatorial and sequential gates, higher-dimensional, multi-valued, complementary, complex computer systems can be built, based on a new logical deep structure, the (TM) Logic/Algebra, and capable of simulating complex systems. These are able to predict complex systems more reliably and comprehensibly. They reduce the susceptibility to errors of existing computer systems.

All known, classical digital architectures can be scaled into these new structures.

Modern Computer Architectures (MCA), such as Dual Rail Computing (DRC), Dual Rail Reversible Computing (DRRC), Quantum Computing (QC), DNA Computing (DNAC), Biocomputing (BC), Neural Networks (NN), Cryptography (KR), can be scaled into the new TM digital structure.

Positive TM Logic/Algebra, like the TM sequence, can be reduced to a mathematical constant 0.0110100….

Negative TM Logic/Algebra is based on the negative TM sequence 10010110… and can theoretically and technically be constructed according to the same principles as positive TM Logic/Algebra.

Through scaling, the classical logic systems are fundamentally extended by recursion, orthogonality, dimensionality, and metasystems.

The recursion principle/scaling principle (principle of recursive definition) is the fundamental construction principle for the theoretically represented (TM) Logic/Algebra.

The recursion principle is based on complete induction. Complete induction is a fundamental and important proof technique in mathematics.

Bits such as (0, 1), (01, 10), or (0110, 1001) are orthogonal to each other. They can be mapped to dimensions 1, 2, 3….

Metasystems are indeed capable of making statements about properties of formal systems such as provability, completeness, consistency, and decidability/undecidability.

In digital technology, they are able to reduce sequential flip-flops/algorithms to combinatorial, algebraic structures.

The (TM) Logic/Algebra can theoretically be used to scale all mathematical structures resulting from classical logic, such as set theory, order structures, algebraic and topological structures of mathematics, into fractal, multidimensional, higher-valued, complementary structures. The aim is to make it possible to reduce and calculate complex "multi-particle systems" based on logic. Conversely, one can start from the current state of science of the (TM) sequence and map its diverse results, characteristics, and applications onto the (TM) Logic/Algebra.

References


[1] Euklid: Die Elemente. Übersetzt und herausgegeben von Clemens Thaer. Wissenschaftliche Buchgesellschaft (WBG), Darmstadt (1980).

[2] Aristoteles: Metaphysik. Reclam Verlag, Ditzingen (1981). (Übersetzung: Franz F. Schwarz; ursprünglich ca. 350 v. Chr.)

[3] Hegel, G. W. F.: Wissenschaft der Logik. Erster Band. Die objektive Logik (1812/13). Herausgegeben von Friedrich Hogemann und Walter Jaeschke. Felix Meiner Verlag, Hamburg (1978)

[4] Boole, G.: An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities. Macmillan, London (1854).

[5] Frege, G.: Begriffsschrift: Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Nebert, Halle a. d. Saale (1879).

[6] Whitehead, A. N.: Russell, B.: Principia Mathematica. Suhrkamp (1986).

[7] Huntington, E. V.: “Postulates for the Algebra of Logic”. Transactions of the American Mathematical Society (1904).

[8] Eccles, W.; Jordan F.: Improvements in Ionic Relays, Patent GB148582, (1918)

[9] Schönfinkel, M.: Über die Bausteine der mathematischen Logik. Mathematische Annalen (1924).

[10] Hilbert, D.; Ackermann, W.: Grundzüge der theoretischen Logik. Springer, Berlin (1928).

[11] Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik (1931).

[12] Birkhoff, G.; von Neumann, J.: The Logic of Quantum Mechanics. Annals of Mathematics (1936).

[13] Weizsäcker, C. F. v.: Komplementarität und Logik. Die Naturwissenschaften 42, (1955).

[14] Weizsäcker, C. F. v.: Komplementarität und Logik II. Zeitschrift für Naturforschung 13a, 245–253 (1958).

[15] Bourbaki, N.: Die Architektur der Mathematik. Teil I & II. In: Physikalische Blätter, Band 17, Nr. 4 & 5, S. 161–166 & 212–218. Wiley-VCH, Weinheim (1961).

[16] Nickel, H.; Birnbaum, H.: Algebra und Geometrie für Ingenieure. 7. Auflage. Fachbuchverlag, Leipzig (1972).

[17] Leupold, W.; Conrad, R.; Völkel, S.; Große, G.; Fucke, R.; Nickel, H.; Mende, H.: Analysis für Ingenieure. Verlag Harri Deutsch, Frankfurt am Main (1975).

[18] Weizsäcker, C. F. v.: Aufbau der Physik. Hanser Verlag, München (1985).

[19] Weizsäcker, C. F. v.: Zeit und Wissen. Hanser Verlag, München (1992)

[20] Weyl, H.: Philosophie der Mathematik und Naturwissenschaft. Oldenbourg Wissenschaftsverlag (2008, 8. Auflage; ursprünglich 1928).

[21] Weyl, H.: Raum – Zeit – Materie: Vorlesungen über allgemeine Relativitätstheorie. Springer-Verlag, Berlin Heidelberg (1988, 7. Auflage).

[22] Shannon, C. E.: A Symbolic Analysis of Relay and Switching Circuits. Transactions of the American Institute of Electrical Engineers, 57(12), 713–723.1 (1938).

[23] Shannon, C. E.: Communication in the Presence of Noise. Proceedings of the IRE, 37(1), 10–21. (1949).

[24] Shannon, C. E.: A Mathematical Theory of Communication. The Bell System Technical J ournal, 27(3), 379–423 und 27(4), 623–656. (1948).

[25] H. Lipp, J. Becker.: Grundlagen der Digitaltechnik, Oldenbourg Verlag, München, (2008).

[26] Pernards Peter.: Digitaltechnik, 3. Auflage, Heidelberg-Hüthig, (1992).

[27] Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; et al.: Scaling Laws for Neural Language Models. arXiv (2020).

[28] Henighan, T.; Kaplan, J.; Katz, M.; et al.: Scaling Laws for Autoregressive Generative Modeling. arXiv (2020).

[29] Aolita, L.; Chaves, R.; Cavalcanti, D.; Acín, A.; Davidovich, L.: Scaling Laws for the Decay of Multiqubit Entanglement. Physical Review Letters (2008).

[30] Franke, D. P.; Clarke, J. S.; Vandersypen, L. M. K.; Veldhorst, M.: Rent's Rule and Extensibility in Quantum Computing. arXiv (2018).

[31] Kolář, M.; Ali, M. K.; Nori, F.: Generalized Thue-Morse Chains and Their Physical Properties. Physical Review B, 43, 1034–1046 (1991).

[32] Carpena, P.; Gasparian, V.; Ortuño, M.: Energy Spectra and Level Statistics of Fibonacci and Thue-Morse Chains. Physical Review B, 51, 12813–12820 (1995).

[33] Bellissard, J.: Spectral Properties of Schrödinger's Operator with a Thue-Morse Potential. In: Luck, J.M., Moussa, P., Waldschmidt, M. (Hrsg.): Number Theory and Physics, Springer, Berlin, Heidelberg (1990).

[34] Meenakshisundaram, N.; Lakshminarayan, A.: The Fourier Transform of the Hadamard Transform: Multifractals, Sequences and Quantum Chaos. arXiv, Cornell University (2005).

[35] N. Meenakshisundaram and A. Lakshminarayan.: Multifractal eigenstates of quantum chaos and the Thue-Morse sequence, Phys. Rev. E 71 065303(R) (2005); J. Phys. A: Math. Gen. bf 39, 11205 (2006).

[36] Fillman, J.: Resolvent Methods for Quantum Walks with an Application to a Thue-Morse Quantum Walk. arXiv, (2017).

[37] Janković, D.; Pasquier, R.; Hartmann, J.-G.; Hervieux, P.-A.: Elucidating the Physical and Mathematical Properties of the Prouhet-Thue-Morse Sequence in Quantum Computing. arXiv, (2025).

[38] Tahir Shah.: The RNA world, automatic sequences, and Oncogenetics, World Laboratory, Lausanne, Switzerland. IC/93/65 INTERNAL REPORT, (1993).

[39] Schat, M.: Using the Thue-Morse Sequence to Cancel Low-Frequency Noise in DNA Microarrays. Proceedings of the IEEE Instrumentation & Measurement Technology Conference (IMTC), 2007.

[40] Li, W.; Thanos, D.; Provata, A.: Quantifying Local Randomness in Human DNA and RNA Sequences Using Erdős Motifs. Journal of Theoretical Biology, Elsevier, 2018.

[41] Yarlagadda, R. K. R.; Hershey, J. E.: The Thue-Morse Sequence. In: Hadamard Matrix Analysis and Synthesis, The Springer International Series in Engineering and Computer Science, vol. 383. Springer, Boston, MA (1997).

[42] Allouche, J.-P.; Peyrière, J.; Wen, Z.-X.; Wen, Z.-Y.: Hankel Determinants of the Thue-Morse Sequence. Annales de l' Institut Fourier, Grenoble (1998).

[43] Allouche, J.-P.; Shallit, J.: The Ubiquitous Prouhet–Thue–Morse Sequence. Springer, Berlin (1999).

[44] Mauduit, C.; Rivat, J.: Multiplicative Properties of the Thue–Morse Sequence. Springer, Berlin (2001).

[45] Spiegelhofer, L.: Thue–Morse Along the Sequence of Cubes. arXiv (2023).

[46] Palacios-Huerta, Ignacio.: Tournaments, Fairness and the Prouhet-Thue-Morse Sequence. Economic Inquiry, Vol. 52, No. 3, pp. 1185-1201 (2014).

[47] Adamczyk, R.: The Thue-Morse Sequence and Boolean Algebra in Music Composition. Selbstverlag, New York (2019).

[48] Gazeau, J.-P.; Lenz, D.; Richard, S.: Abstract Art Generated by Thue–Morse Autocorrelation Functions. Springer, Berlin (2023).

[49] Séébold, Patrice.: Overlaps in purely morphic words. Theoretical Computer Science, Vol. 165, Issue 1, pp. 141-155 (1996).

[50] Brlek, Srečko.: On the combinatorics of words. Discrete Applied Mathematics, Vol. 154, Issue 2, pp. 181-206 (2006).

[51] Baake, M.; Gähler, F.; Grimm, U.: Spectral and Topological Properties of a Family of Generalised Thue-Morse Sequences. Journal of Mathematical Physics, American Institute of Physics, (2012).

[52] Dekking, M.: On the Structure of Thue-Morse Subwords, with an Application to Dynamical Systems. arXiv, (2014).

[53] Peng, L.; Kamae, T.: Spectral Measure of the Thue–Morse Sequence and the Dynamical System and Random Walk Related to It. Ergodic Theory and Dynamical Systems, Cambridge University Press, (2016).

[54] Baake, M.; Coons, M.: Correlations of the Thue-Morse Sequence. Indagationes Mathematicae, Elsevier, (2024).

[55] Wolfgang Hilberg.: Mehrdimensionale Morse-Thue-Folgen, Physik in unserer Zeit / 22. Jahrg. (1991 / Nr. 1).

[56] Alexis De Vos.: Reversible Computing, Fundamentals, Quantum Computing and Application, Wiley-VCH-Weinheim (2010).

[57] De Vos, A.; De Baerdemacker, S.; Van Rentergem, Y.: Synthesis of Quantum Circuits vs. Synthesis of Classical Reversible Circuits. Morgan &

Claypool Publishers, San Rafael, CA (2018).

[58] Hettich, P.: Scalable Circuit Assembly and method for producing a logic AND-gate switching network and method for producing a corresponding computer, WIPO-WO 2025/061229 A1, (2025).