Original Article

A new binary number system for real numbers

Deliang Shi [1] Ⓛ

Abstract

A number system uses a set of digits and sign to represent all the numbers. Different number systems use different amount of digits and sign. To compare the leanness of a number system, the cardinality of the complete set of digits and sign are employed in this work. The 01 binary number system is used by almost all the modern computers. It can represent all the real numbers by two digits 0, 1 and a sign (-), which means its cardinality is 3. Thus, the 01 binary system is not a true binary system. After reviewing all the existing number systems it is found that no true binary system exists for real numbers. All the current binary number systems either don't have radix 2, or have cardinality 3. In this paper, a true binary system for real numbers, with both radix and cardinality 2, is invented based on the principle of Yin-Yang. Surprisingly, this binary system can not only represent all real numbers without using a sign, but also perform arithmetic and Boolean operations without any problem. Furthermore, due to its radix being 2, this true binary system is compatible with the 01 binary system and many operations are interchangeable between the two systems. It is expected that this true binary system will be applied in the fields of number representation, computer science and cryptography.

Key words: Binary number system; Cardinality; Yin-Yang; Real numbers

---

Affiliation Info: [1] Independent Researcher, Vernon Hills, IL 60061, U.S.A.

Author's Contact Info: Shi, DL: deliangshi@gmail.com

Citation: Deliang Shi. 2024. A new binary number system for real numbers. *Naturalis Scientias*, 1 (4): 286-302. DOI: https://doi.org/10.62252/NSS.2024.1020.

Corresponding Aurhor Ⓛ: DL Shi, PhD; Email: deliangshi@gmail.com

1. Introduction

There are many different positional number systems. The conventional decimal number system uses 10 digits 0,1…9 to represent numbers and a sign to indicate if it is positive or negative, such as,

$$-987.12 = -(9 \times 10^2 + 8 \times 10^1 + 7 \times 10^0 + 1 \times 10^{-1} + 2 \times 10^{-2}).$$

The 01 binary system is a base-2 system. It can represent every real number using 2 digits and a sign. This system is widely used in the modern computer science[1-6], such as,

$$-(110.01)_2 = -(1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}).$$

The sign has caused huge complexity on representing numbers in the modern computer system, since the modern computer system relies on two-stage, on and off, to represent all the number. In order to represent both positive and negative numbers, several methods have been used, such as signed magnitude, two's complement, one's complement notations etc. Unfortunately, each method has its own advantages and disadvantages[7].

There are some number systems that don't need sign inherently, as illustrated in below generalized numeral system.

A generalized numeral system in base b can be expressed as

$$(\dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots)_b = a_2 \times b^2 + a_1 \times b^1 + a_0 \times b^0 + a_{-1} \times b^{-1} + a_{-2} \times b^{-2},$$

where b can be integers, non-integers, positive numbers, negative numbers, irrational numbers and complex numbers etc. In the generalized numeral system, the sign is not required for every number system, since both positive and negative numbers can be represented without sign in a negative based system, such as,

$$(1011)_{-2} = 1 \times (-2)^3 + 0 \times (-2)^2 + 1 \times (-2)^1 + 1 \times (-2)^0 = -9.$$

The complex-base number systems don't need sign as well due to that negative number can be obtained by the multiplication of imaginary numbers. A "binary" complex system based on $\pm\sqrt{2}i$ can represent every complex number using only 0 and 1[8]. Another "binary" complex number system based on $i - 1$ can also represent every complex number using only 0 and 1[9-10], such as,

$$(101)_{i-1} = 1 \times (i-1)^2 + 0 \times (i-1)^1 + 1 \times (i-1)^0 = 1 - 2i,$$

Unfortunately, both the negative and complex based number systems are not based on positive integers, thus, their applications are very limited.

The balanced ternary system is a base-3 system. It can represent every real number using $-1, 0, 1$ without a sign and some early computers were developed based on this system[11]. The balanced binary system is a base-2 system. It can represent every real number using -1, 0, 1 without a sign[12-13]. The drawback of these two balanced number systems is that they still need 3 symbols to represent real numbers, which is the same amount as the 0, 1 and sign binary system.

In summary, a lot of efforts have been put on getting a positive integer based system without sign. However, to represent real numbers, all the existing positive integer based systems need use at least 3 symbols, which doesn't match the two-stage of a computer system. There are some negative and complex based number system that can represent all numbers using 2 symbols, but their radixes are not 2 and they are not convenient on general usage. Thus, there exists no true binary system for real numbers.
In this work, a true binary system that can represent real numbers using just two symbols without sign will be demonstrated.

2. Cardinality of a positional number system

A positional number system is represented as

$$\langle \rho, Z \rangle$$

with radix ρ and set of digits Z, possibly with a sign $\pm$[14]. For example, the common 01 binary system has

$$\rho = 2$$

and

$$Z = \{0,1\}.$$

It is represented as

$$\langle 2, \{0,1\} \rangle.$$

Obviously, without the sign, $\langle 2, \{0,1\} \rangle$ can only represent non-negative numbers, i.e.

$$\langle 2, \{0,1\} \rangle \text{ for any } x \in R_+,$$

where R represents the domain for real numbers and R_+ represents the domain for non-negative real numbers.
To represent all real numbers, the sign is required. Here we define Z_c as the complete set of symbols of a numeral system, which include the necessary sign as well.

$$Z_c = \begin{cases} Z & \text{if the system doesn't need sign;} \\ Z \cup \{\pm\} & \text{if the system need sign.} \end{cases}$$

Thus, the positional number system pair becomes

$$\langle \rho, Z_c \rangle.$$

In above notation, there is no need to say if the system need a sign or not, since Z_c is the complete set of all symbols.
Here, we define the cardinality of a positional number system to be the number of elements in set Z_c, i.e.,

$$card(\langle \rho, Z_c \rangle) = |Z_c|.$$

For binary numeral system representing non-negative numbers , we have

$$Z_c = Z = \{0,1\}.$$

For binary numeral system representing real numbers, we have

$$Z_c = Z \cup \{\pm\} = \{0,1,\pm\}.$$

Thus, we have

$\langle 2, \{0,1,\pm\}\rangle$ for any $x \in R$.

In above notation, the number system and its domain have been shown together, i.e. in below general form,

$$\langle \rho, Z_c \rangle \; for \; any \; x \in D \; .$$

Since a number system is always related to a domain, we will use above notation to clearly show a number system and its domain.

The cardinality of a number system can be calculated, such as

$$|Z_c| = |\{0,1,\pm\}| = 3,$$

i.e. there are 3 symbols in the number system $\langle 2, \{0,1,\pm\}\rangle$ for any $x \in R$.

Similarly, the balanced binary system can be represented as

$\langle 2, \{-1, 0, 1\}\rangle$ for any $x \in R$.

We have

$$|Z_c| = |\{-1,0,1\}| = 3,$$

i.e. there are 3 symbols in the number system $\langle 2, \{-1, 0,1\}\rangle$ for any $x \in R$.

The cardinality of a positional number system can be used to compare different number systems. The smaller the cardinality, the leaner the system. For any $\in D$, if there are two number systems $\langle \rho_1, Z_{c1}\rangle$ and $\langle \rho_2, Z_{c2}\rangle$, we can compare them as below,

if $|\mathrm{Z_{c1}}| = |\mathrm{Z_{c2}}|$, then $\langle \rho_1, \mathrm{Z_{c1}}\rangle$ is as lean as $\langle \rho_2, \mathrm{Z_{c2}}\rangle$;

if $|\mathrm{Z_{c1}}| < |\mathrm{Z_{c2}}|$, then $\langle \rho_1, \mathrm{Z_{c1}}\rangle$ is leaner than $\langle \rho_2, \mathrm{Z_{c2}}\rangle$.

If a number system has cardinality $|Z_c|$, it is possible to get a new number system which has a bigger cardinality than $|Z_c|$ for that domain by adding some redundancy into the new number system. Though, it is not always true vice versa, since it is not always possible to make a number system leaner if that number system is already the leanest one in the field. Thus, finding the leanest system for a specific field is critical to obtain other possible number systems.

It is easy to see number systems

$\langle 2, \{0,1,\pm\}\rangle$ for any $x \in R$

and

$$\langle 2, \{-1, 0, 1\}\rangle \text{ for any } x \in R$$

are applied in the same field, i.e. $x \in R$ and they have the same value of $|Z_c|$, i.e. 3. Thus, these two numeral systems are as lean as each other.

It is not hard to check and find that, for $\in R$, all the existing positive integer based number systems having $|Z_c| \geq 3$.

The concept of cardinality can be applied to the computer system as well, since Z_c is defined as the complete set of symbols, not only digits. The cardinality of the computer system can be calculated as,

$$|Z_c| = |\{on, off\}| = 2.$$

Thus, we can clearly see that the gap between the existing binary system and the computer system. The computer system is true binary, since it has only 2 symbols though the existing binary system has 3 symbols. These two systems are not bijective. To close this gap, there are two possible routes.

Route 1: Bring up the $|Z_c|$ of the computer system to 3, to match the number system. There are mainly 3 solutions in this route.

1) Use a 3-state computer such as

$$Z_c = \{\text{off}, \text{low}, \text{high}\}$$

or

$$Z_c = \{\text{negative}, \text{off}, \text{positive}\}$$

2) Define an extra sign bit

$$Z_c = \{on, off, sign\}.$$

3) Use the combination of on and off to form 3 symbols such as in the redundancy binary representation[15-16], i.e. ,

$$Z_c = \{on - on, on - off, off - off\}.$$

It is not hard to see that Route 1 is making the whole system more complicated. Unfortunately, most of the existing solutions are taking this route.

Route 2: Find a new binary number system $\langle 2, Z_c\rangle$ for any $x \in R$ with $|Z_c| = 2$, to match the computer system.

Obviously, Route 2 is reducing the cardinality to simplify the system. So, it is more preferred than route 1. The challenge is that a new binary system need be found and work well, if this new binary system ever exists.

In this work, I will take Route 2 to close the gap. I will use a new way to obtain a true binary system which meet below requirements,

$$\langle 2, Z_c\rangle \text{ for any } x \in R \text{ with } |Z_c| = 2.$$

Obviously, this true binary system will become the leanest base-2 binary system for real numbers. In the end of this work, you will understand why the traditional well-established number system theories don't really help here. Sometimes they are even misleading on finding this true binary number system, such as it is mentioned that there are only two systems of binary coding, namely $\langle 2, \{0, 1\}\rangle$ and $\langle -2, \{0, 1\}\rangle$[14].
Before moving forward, let's jump out of the box for the time being and goes back to the origin of the modern binary number system.

3. Binary and Yin-Yang

The modern binary numeral system was invented by Gottfried Wilhelm Leibniz[17]. In his work, Leibniz noticed the similarity between Yin-Yang and the 01 binary system[17-18]. In the Yin-Yang And Eight Trigrams Graph, Yin is represented as a broken line - - and Yang is represented as a whole line —. Combinations of Yin and Yang comprise the Eight Trigrams ☷, ☳, ☵, ☲, ☶, ☴, ☱, ☰.
Replacing Yin with 0 and Yang with 1, Leibniz noticed that the eight trigrams will become 000, 001,010, 011, 100, 101, 110, 111, as shown in Figure 1.

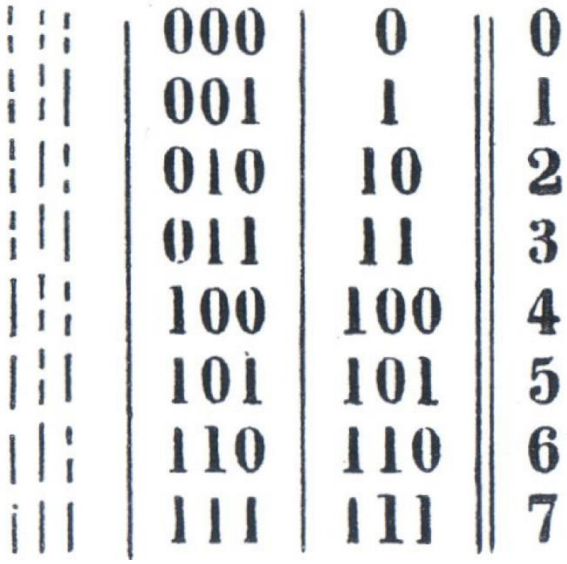

Figure 1. The similarity between the 01 binary and Yin-Yang noticed by Leibniz[17]

4. Yin-Yang binary number system

After reviewing Leibniz's work, one has to point out that, even if the 01 binary system is similar to Yin-Yang, the 01 representation cannot reflect the true meaning of Yin-Yang. In Yin-Yang theory, Yin means negative and Yang means positive[19-24]. Yin and Yang are equal but opposite. They are complementary to each other and can make up everything without resorting to a third

party. Using 0 to represent Yin cannot show that Yin is opposite to Yang. In fact, 0 represents neutral instead of negative. Based on the Yin-Yang theory, it is better to use -1 and 1 to represent Yin and Yang than to use 0 and 1.

Thus, a true binary system, using -1 and 1 as the digits, is invented in this work, i.e.

$$\langle 2, \{-1, 1\}\rangle \text{ for any } x \in R.$$

Surprisingly, this binary system works very well on number representation, as will be demonstrated in this work.

Since this true binary system is based on the principle of Yin-Yang, we will call it Yin-Yang binary system in this work. We will use $\bar{1}$ to represent -1, thus we have

$$Yin = -1 = \bar{1},$$

and

$$Yang = 1.$$

The Yin-Yang binary system will be shown as

$$\langle 2, \{\bar{1}, 1\}\rangle \text{ for any } x \in R.$$

The decimal value of a number in Yin-Yang binary system can be calculated, such as,

$$\bar{1}\bar{1} = (-1) \times 2^1 + (-1) \times 2^0 = -2 - 1 = -3.$$

It is obvious that -3 is negative and it has been represented in the Yin-Yang binary system without using the negative sign.

The decimal value of a fraction number can be calculated as well, such as,

$$11.\bar{1}1 = 1 \times 2^1 + 1 \times 2^0 + (-1) \times 2^{-1} + 1 \times 2^{-2} = 2 + 1 - 0.5 + 0.25 = 2.75.$$

A repeating number can also be calculated, such as,

$$.\dot{1} = .111\cdots = 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + \cdots = \sum_{n=1}^{\infty} \frac{1}{2^n} = 1,$$

where the dot (·) above the number indicates a repeating number.

Similarly, we can calculate that below numbers are all equal to 0. Thus, 0 can be represented in the Yin-Yang binary system even though it is not included in the basic digits,

$$\bar{1}.\dot{1} = 1.\dot{\bar{1}} =. \bar{1}\dot{1} = .1\dot{\bar{1}} = 0.$$

It is different from other non-zero positional number systems, such as in the bijective base systems, an empty string is used to represent 0[25].

A real number x in decimal can be converted to y in the new Yin-Yang binary system by following below procedure.

If x<0, y =-1.

Otherwise, y=1.

p=floor($\log_2|x|$).

$x=x-2^p$.

Then, repeat below process until accuracy is met.

If $x<0$, $y=-1$.

Otherwise, $y=1$.

$p=p-1$.

$x=x-2^p$.

The sequence of y gotten from this procedure will be the digits in the Yin-Yang binary system. Table 1 shows an example of converting the decimal number 99 to Yin-Yang binary.

Table 1. Conversion of number in decimal to Yin-Yang binary

X	Y	$\log_2\|x\|$	position	$2^{position}$	$x-y\times2^{position}$
99	1	64	6	$2^6=64$	35
35	1		5	$2^5=32$	3
3	1		4	$2^4=16$	-13
-13	-1		3	$2^3=8$	-5
-5	-1		2	$2^2=4$	-1
-1	-1		1	$2^1=2$	1
1	1		0	$2^0=1$	0

Based on the column of y, 99 in decimal is converted to $11\bar{1}\bar{1}\bar{1}1$ in the Yin-Yang binary system.

Please refer to the Supplement Information for a code written in C language that can convert any real number to Yin-Yang binary. Some typical numbers represented in different numeral systems are shown in Table 2.

Table 2. Some typical numbers represented in different numeral systems

Decimal Need 11 symbols (0,1,2…9 and -) to represent all the real numbers	**01 Binary** Need 3 symbols (0,1 and -) to represent all the real numbers	**Yin-Yang Binary** Need only 2 symbols ($\bar{1}$ and 1) or (-1 and 1) to represent all the real numbers
0	0	$\bar{1}.\dot{1}$
1	1	1
2	10	$11.\dot{\bar{1}}$
3	11	11
4	100	$11.\dot{1}$
5	101	$11\bar{1}$
6	110	$111.\dot{\bar{1}}$
7	111	111
8	1000	$1\bar{1}11.\dot{1}$
9	1001	$11\bar{1}\bar{1}$
10	1010	$11\bar{1}1.\dot{\bar{1}}$
-1	-1	$\bar{1}$
-2	-10	$\bar{1}\bar{1}.\dot{1}$
-3	-11	$\bar{1}\bar{1}$
-4	-100	$\bar{1}\bar{1}.\dot{\bar{1}}$
-5	-101	$\bar{1}\bar{1}1$
-6	-110	$\bar{1}\bar{1}\bar{1}.\dot{1}$
-7	-111	$\bar{1}\bar{1}\bar{1}$
-8	-1000	$\bar{1}\bar{1}\bar{1}.\dot{\bar{1}}$

-9	-1001	$\bar{1}\bar{1}11$
-10	-1010	$\bar{1}\bar{1}1\bar{1}.\dot{1}$
0.1	$0.0\overline{0011}$	$.1\dot{\bar{1}}\dot{\bar{1}}\dot{1}\dot{1}$
0.2	$0.\overline{0011}$	$.1\dot{\bar{1}}\dot{\bar{1}}\dot{1}\dot{1}$
0.25	0.01	$.1\bar{1}$
1/3	$0.\overline{01}$	$.\dot{1}\dot{\bar{1}}$
0.5	0.1	$.1$
π (3.14159265…)	11.001001…	$11.1\bar{1}\bar{1}1\bar{1}\bar{1}…$
8848.13	10001010010000.001	$11\bar{1}\bar{1}1\bar{1}1\bar{1}\bar{1}1\bar{1}\bar{1}1.\bar{1}\bar{1}\bar{1}$
-430.5	-110101110.1	$\bar{1}\bar{1}\bar{1}1\bar{1}1\bar{1}\bar{1}1.1$

5. Expandability

In the Yin-Yang binary system, we can find below relationship

$$\bar{1} = \bar{1}1 \text{ and } 1 = 1\bar{1}.$$

This relationship is obvious, since

$$-1 = -2 + 1 \text{ and } 1 = 2 - 1.$$

Furthermore, above relationship can be expanded into multiple digits by replacing the leftmost symbol with two symbols in the right side of the above equations, such as

$$\bar{1} = \bar{1}1 = \bar{1}11 = \bar{1}111111 ...,$$

and

$$1 = 1\bar{1} = 1\bar{1}\bar{1} = 1\bar{1}\bar{1}\bar{1}\bar{1}\bar{1} ....$$

Above relationship can be easily proven since $2^n = 2^{n+1} - 2^n$ (for any $n \in R$). The expandability of the Yin-Yang numbers is useful in the arithmetic operations, which will be discussed in the following content.

6. Arithmetic operations

The Yin-Yang binary numbers can perform all the arithmetic operations. It is easy to see below addition operations are valid.

$$1 + 1 = 1.\dot{1}$$

$$\bar{1} + \bar{1} = \bar{1}.\dot{\bar{1}}$$

$$1 + \bar{1} = 1.\dot{\bar{1}}$$

$$1 + 1 + 1 = 11$$

$$\bar{1} + \bar{1} + \bar{1} = \bar{1}\bar{1}$$

$$1 + 1 + \bar{1} = 1$$

$$1 + \bar{1} + \bar{1} = \bar{1}$$

From above addition operations, we can find the addition of 2 digits in Yin-Yang binary will still yield 2 digits. But, the addition of 3 digits will yield only 1 or 2 digits. Thus, as a thumb of rule, we will change the addition of 2 digits into the addition of 1 or 3 digits, by employing the expandability, as exemplified in Table 3.

Since subtraction is inverse to addition, multiplication is repeated addition, and division is inverse to multiplication, we can expect these operations will be successfully performed if the addition operation can be successfully performed.

The three basic subtractions are:

$$\bar{1} - \bar{1} = \bar{1} + 1 = \bar{1}.\dot{1}$$

$$1 - 1 = 1 + \bar{1} = \bar{1}.\dot{1}$$

$$1 - \bar{1} = 1 + 1 = 1.\dot{1}$$

The three basic multiplication operations are:

$$1 \times 1 = 1$$

$$\bar{1} \times \bar{1} = 1$$

$$1 \times \bar{1} = \bar{1}$$

The three basic division operations are:

$$1/1 = 1$$

$$\bar{1}/1 = \bar{1}$$

$$1/\bar{1} = \bar{1}$$

Examples of general arithmetic operations are shown in Table 3.

Table 3. Addition, subtraction, multiplication and division operations in Yin-Yang binary

Addition	**Subtraction**	**Multiplication**	**Division**
$\bar{1}\ \bar{1}\ 1$ -5 $+\quad 1\ 1$ 3 $\bar{1}\ \bar{1}\ 1\ .\ \dot{1}$ $+\quad 1$ $\bar{1}\ \bar{1}\ \bar{1}\ .\ \dot{1}$ 1 $+\quad 1$ $\bar{1}\ 1\ \bar{1}\ .\ \dot{1}$ -2 $\bar{1}\ \bar{1}\ .\ \dot{1}$ -2	$\bar{1}\ \bar{1}\ 1$ -5 $-\quad 1\ 1$ 3 $\bar{1}\ \bar{1}\ 1\ .\ \dot{1}$ $-\quad 1$ $\bar{1}\ \bar{1}\ \bar{1}\ .\ \dot{1}$ 1 $-\quad 1$ $\bar{1}\ \bar{1}\ \bar{1}\ .\ \dot{1}$ -8	$\bar{1}\ \bar{1}\ 1$ -5 $\times\quad 1\ 1$ 3 $\bar{1}\ \bar{1}\ 1$ $+\ \bar{1}\ \bar{1}\ 1$ $\bar{1}\ \bar{1}\ \bar{1}$ $\bar{1}\ \bar{1}\ 1$ $+\quad 1$ $\bar{1}\ 1\ \bar{1}$ $+\ \bar{1}\ \bar{1}$ $\bar{1}\ \bar{1}\ \bar{1}$ $\bar{1}\ \bar{1}$ $+\quad 1$ $\bar{1}\ \bar{1}\ \bar{1}\ \bar{1}$ -15	$\bar{1}\ 1$ -1 $3 \mid 1\ 1 \overline{)\ \bar{1}\ \bar{1}\ 1}$ -5 $-\ \bar{1}\ \bar{1}$ 1 $-\quad 1\ 1$ $\bar{1}\ 1\ .\ \dot{1}$ -2

The operations are illustrated by -5+3, -5-3, -5×3 and -5/3.

7. Boolean operations

Boolean operations can also be performed in Yin-Yang binary. There are 3 basic Boolean operations, i.e. AND, OR and NOT. More complicated Boolean algebra can be achieved by combing these three basic Boolean operations[26].

AND is defined as X AND Y=1 if X=Y=1, otherwise X AND Y=$\bar{1}$.

Here, X AND Y is similar to minimum operation, i.e., X AND Y =min(X, Y).

OR is defined as X OR Y=$\bar{1}$ if X=Y=$\bar{1}$, otherwise X OR Y=1.

Here, X OR Y is similar to the maximum operation, i.e. X OR Y=max(X, Y)

NOT is defined as NOT X=$\bar{1}$ if X=1 and NOT X=1 if X=$\bar{1}$.

Here, NOT X is similar to the negative operation, i.e. -X.

We can see the AND and OR operations are similar to those in 01 binary system. The NOT operation is the negative operation and it is more instinct than that in 01 binary system.
The Truth Table is summarized in Table 4.

Table 4. The Truth Table in Yin-Yang binary

X	Y	X and Y	X or Y
$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$
1	$\bar{1}$	$\bar{1}$	1
$\bar{1}$	1	$\bar{1}$	1
1	1	1	1

X	not X
$\bar{1}$	1
1	$\bar{1}$

It is noted that the Boolean Operations in Yin-Yang binary can be easily converted to Zadeh Operation for Fuzzy logic[27-28], except that the Not operation should also be the Negative Operation, as shown in Table 5.

Table 5. Fuzzy logic operations in Yin-Yang binary.

Boolean Operations	Fuzzy Logic Operations
X AND Y	min(X, Y)
X OR Y	max(X, Y)
NOT X	-X

8. Discussion

So far, we have obtained a true binary number system which meets the requirement

$$\langle 2, Z_c \rangle \text{ for any } x \in R \text{ with } |Z_c| = 2,$$

i.e., the Yin-Yang binary system

$$\langle 2, \{\bar{1}, 1\} \rangle = \langle 2, \{-1,1\} \rangle \text{ for any } x \in R.$$

This number system can represent every real number without sign. Furthermore, it can perform all the arithmetic and Boolean operations easily. Besides that, it also has below advantages:

1) The negative of a number can be obtained by exchanging 1 and -1.
2) The sign of a number is given by its most significant digit, except 0, which can be either positive or negative.

3) Since the system is based on 2, it is compatible with the current 01 binary system when performing arithmetic operations. Thus, many operations could be performed seamlessly between the two systems, as shown in Table 6.

Table 6. Arithmetic operations between Yin-Yang Binary and 01 binary are compatible

	$\bar{1}$	1	Yin-Yang Binary
+	1	0	01 Binary
		1	Result

The disadvantage of this number system is that it need use infinite repeating fraction to represent even integers. But, as most of real numbers are approximated in the 01 binary system, the accuracy of any number in the Yin-Yang binary system can be represented accurate enough and it is only limited to the capacity of the computer.

Here let's discuss why this true binary number system can only be obtained by a non-traditional way. In traditional theories of number systems, there are many stereotypes, such as most number systems include 0 in the system, or the digits are continuous integers. For example, the digits of a balanced number system base-b is generalized as[29],

$$-k\ to\ (b-1)-k \text{ where } \quad k=\left\lfloor\frac{b}{2}\right\rfloor.$$

We can see, as a balanced number system, Yin-Yang binary system doesn't follow above generalized rule. Thus, it may explain why it was not found before.

As a true binary system for representing real number, Yin-Yang binary system have a lot of potential applications. In many cases there are only two symbols available, such as a coin having only two sides, a magnet having only two poles and a black white printer can only show two colors. Thus, using only 2 symbols to represent as much information as possible is often not only preferred, but also required[30-31]. Yin-Yang binary system provides the mathematical foundation for using only 2 symbols in a base-2 system to represent all real numbers, which has never been achieved before by any other number systems.

9. Conclusions

The cardinality of the complete set of digits and sign is defined in this work to compare the leanness of different number systems. It is found that the existing binary systems all having cardinality 3, thus they are not true binary systems.

A true binary number system based on the principle of Yin-Yang is introduced in this work. Comparing to the existing binary systems, Yin-Yang binary system has cardinality 2 and uses only 2 symbols, thus it is a true binary system. It can not only represent all real numbers, but also perform arithmetic and Boolean operations without any problem.

10. References

1. Randell B. 1982. From analytical engine to electronic digital computer: the contributions of Ludgate, Torres, and Bush. *IEEE Ann. Hist. Comput*., 4 (4) : 327–341.

2. Zuse K. 1993. *The Computer - My Life*. Springer Science & Business Media, Berlin .

3. Rojas R. 1997. Konrad Zuse's Legacy: The Architecture of the Z1 and Z3. *IEEE Ann. Hist. Comput*., 19 (2): 5–15.

4. Copeland B. 2017. "The modern history of computing" in the *Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University.

5. Shannon C. & Weaver W. 1949. *The Mathematical Theory of Communication*. University of Illinois Press.

6. Stokes J. 2007. *Inside the machine: an illustrated introduction to microprocessors and computer architecture*. No Starch Press, San Francisco.

7. Knuth D. 1997. "Seminumerical Algorithms" in the *Art of Computer Programming*. Addison-Wesley, 2.

8. Knuth D. 1960. An imaginary number system. *Commun. ACM*., 3 (4): 245–247.

9. Khmelnik S. 1962. "A Specialized Computer for Operations on Complex Numbers" in *Questions of Radio Electronics* XII, No. 2.

10. Penney W. 1965. A "binary" system for complex numbers. *JACM.*, 2 (12): 247-248.

11. Glusker D, Hogan P & Vass. 2005. The ternary calculating machine of Thomas fowler. *IEEE Ann. Hist. Comput*., 3 (27): 4–22.

12. Reitwiesner G. 1960. Binary arithmetic. *Adv. in Compu.*, 1: 231-308.

13. Knuth D. 1986. Efficient balanced codes. *IEEE Trans. Inf. Theory.*, 1 (32): 51–53.

14. Khmelnik S & Doubson I. 2011. Positional codes of complex numbers and vectors. *Mathematics in Computers Corp.*, Israel.

15. Lapointe M, Huynh H & Fortier P. 1993. Systematic design of pipelined recursive filters. *IEEE Trans Comput.*, 4 (42): 413–426.

16. Phatak D & Koren I. 1994. Hybrid signed-digit number systems: A unified framework for redundant number representations with bounded carry propagation chains. *IEEE Trans Comput.*, 8 (43): 880–891.

17. Leibniz G. 1879. Explanation of binary arithmetic, which uses only the characters 1 and 0, with some remarks on its usefulness, and on the light it throws on the ancient Chinese figures of Fu Xi. *Mathematical Writings VII.* ed. C. Gerhardt, Berlin, 7: 223.

18. Ryan J. 1996. Leibniz' binary system and Shao Yong's "Yijing". *Philos. East West.*, 1 (46): 59–90.

19. Wang H. (Ed.). 2011. *I Ching*. Yunnan People's Publishing House, Kunming.

20. Richard W, Baynes C & Brown J. 1977. *The I Ching or Book of Changes*. Princeton University Press.

21. Needham J. 1956. Science and civilization in China. 2: *History of Scientific Thought*. Cambridge University Press.

22. Zhang W, Pandurangi A &Peace K. 2007. Yin Yang dynamic neurobiological modeling and diagnostic analysis of major depressive and bipolar disorders. *IEEE Trans. Biomed. Eng.*, 10 (54): 1729-1739.

23. Hahn S. 1992. The Yin and the Yang of mammalian transcription. *Curr. Biol.*, 3 (2): 152-154.

24. Gore J & Oudenaarden A. 2009. The yin and yang of nature. *Nature*, 457: 271–272.

25. Foster J. 1947. A number system without a zero symbol, *Mathematics Magazine*, 1 (21): 39-41.

26. Givant S & Halmos P. 2009. *Introduction to Boolean Algebras*. Springer, New York.

27. Zadeh L. 1965. Fuzzy sets. *Information and Control*, 3 (8): 338-353.

28. Novak V, Perfilieva I & Mockor J. 1999. *Mathematical Principles of Fuzzy Logic*. Springer US.

29. Parhami B. 1988. Carry-free addition of recoded binary signed-digit numbers. *IEEE Trans. Comput.*, 11 (37): 1470-1476.

30. Hankerson D, Menezes A & Vanstone S. 2004. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, New York.

31. Querini M, Grillo A, Lentini A & Italiano G. 2011. 2D color barcodes for mobile phones. *Int. J. Compu. Sci. App.*, 1 (8): 136-155.

Acknowledgements

I want to thank F.T. Shi for teaching me the knowledge of Yin-Yang. I want to thank Dr. X. Fan for reviewing the initial manuscript.

Data availability

The data that support the findings of this study is available from the author upon reasonable request.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Use of AI tools declaration

The author declares that he has not used Artificial Intelligence (AI) tools in the creation of this article.

Appendix:

Below is a code written in C language that can convert any real number to the Yin-Yang binary number system.

```
/***************** convert_decimal_to_yinyang.c ***************/
#include <stdio.h>
#include <math.h>

float x; /*Any real number*/
int y;   /*Yin-Yang binary*/
int p;   /*Digit position in Yin-Yang binary */

int main() {
  printf("Please input any real number, then press carriage return.\n");
  scanf("%f",&x);
  printf("\n%f in decimal is \n", x);

  y=1;
```

```
 if(x<0) y=-1;
 p=floor(log2(fabs(x)));
 if(fabs(x)<1) {
  printf(". ");
  p=-1;
 }
 printf("%d ",y);

while(p>-10){
 x=x-y*pow(2,p);
 if(p==0) {
   if(fabs(x)<1e-6) break;
   printf(". ");
 }
 y=1;
 if(x<0) y=-1;
 printf("%d ",y);
 p=p-1;
}
 printf("\nin Yin-Yang binary.\n");
 return 0;
}
/********************End of the Code*****************************/
```