# Structural constructions of computer engineering elements in k-valued logic

© Volodymyr Yartsev [1] and © Dmitry Gololobov [2]

[1] State University of Telecommunications, Kyiv, Ukraine
[2] National Aviation University, Kyiv, Ukraine
jvp57@ukr.net, gololobov.dma@meta.ua

**Abstract.** Methods are proposed for implementing the logical functions of digital devices without memory and with memory using three-valued logic obtained using direct and inverse multiplexer circuits as a logical basis. This simplifies the development of digital logic circuits with a large number of input arguments, reduces the number of tiers of the tree structure of the circuit implementation. Using a multiplexer to perform operations in three-valued logic allows reducing the number of logical elements in the structure of a digital device, reducing the delay time when generating a control signal, and using existing microcircuits with transistor-transistor logic. As an example, structural synthesis of a finite state machine recognizing a given combination is given.

**Keywords:** digital devices, logical basis, finite-state machine, logic function, multiplexer, ternary adder, ternary trigger.

## 1 Introduction

In modern discrete mathematics and the theory of building digital technology, an important place is occupied by the problem of building state machine based on the conversion of Boolean functions. Various digital devices on which artificial intelligence systems, control systems are built, solve complex computational problems based on elementary binary operations. But today, the physical and technological parameters of binary logic microchips have reached the practical limit in both size and speed. Further growth of these indicators is associated with the use of calculators that work in multi-valued alphabets [1].

The complexity of the tasks that digital devices perform is constantly growing. This requires an increase in the operating frequency, the number of logic elements in the circuits, and at the same time, a reduction in the cost of power consumption. The use of multi-valued logic is one way to resolve these problems. The multi-valued logic provides more opportunities for the development of various information processing algorithms, which allows to reduce the computational complexity, the number of elements and connections in various signal conditioning devices, increase the density of elements, increase the speed and volume of processed data [2].

Professor of Stanford University D. Knut in his book "The Art of Programming" emphasized that ternary logic is more elegant and more efficient than binary, and in the future, humanity will again return to its development [3]. The multi-valued logic has been widely used to solve the problems of data transmission and storage, the formation of a visual display of graphic information and the processing of complex digital signals [4].

The following conclusions can be drawn from the analysis of literary sources. Most of the technical solutions of the sets of logic and operating devices use a symmetric triple system with a set of signals corresponding to the values of –1, 0, +1, which are technically implemented on the threshold elements of three-valued logic in integrated electronics [5- 6].

## 2 Problem Formulation

In this paper, the theoretical and practical problems of the implementation of digital devices, in particular, the finite state machine, in ternary number system are considered. Digital devices in this number system are the simplest in technical implementation compared to other positional number systems with a base of more than two [7]. In addition, ternary computers are considered promising for the creation of artificial intelligence, since ternary logic is the most natural way of thinking from the point of view of human thinking, and the cognition process, along with the categories "yes" and "no" to describe the state of uncertainty, the meaning of "unknown" or "may be". Note that three-valued logic is considered as a possible replacement for binary logic in the development of quantum computers [8–9]. To construct structural schemes in any multi-valued logic, it is necessary to have a functionally complete logical basis and a memory element with a functionally complete system of transitions, states, and outputs. [10] The study of these issues is focused on in this paper.

## 3 Choosing a logical basis

The total number of N logic functions in three-valued logic is:

$$N = 3^{3^r}$$

where r is the argument of a logical function. The properties of many logical functions of multi-valued logics are still not well understood, but at the moment, among the many logical functions of each k-valued logic, two can be distinguished, which, together with the constants 0, 1, 2, ..., k-1, create two universal logical bases. These logical functions, as well as the devices that implement them (logic elements), are called forward and reverse multiplexers.

The forward multiplexer (**Помилка! Джерело посилання не знайдено.**, left) function has the form:

$$M_{x_0}^{k}(x_1, x_2, ..., x_k) = \begin{cases} x_1 \quad if \quad x_0 = 0 \\ x_2 \quad if \quad x_0 = 1 \\ \text{..........................} \\ x_k \quad if \quad x_0 = k\text{-}1 \end{cases}$$

The reverse multiplexer (Fig. 1, right) function is written as follows:

$$\overline{M}_{x_0}^{k}(x_k, x_{k-1}, ..., x_1) = \begin{cases} x_k \quad if \quad x_0 = 0 \\ x_{k-1} \quad if \quad x_0 = 1 \\ \text{..........................} \\ x_1 \quad if \quad x_0 = k\text{-}1 \end{cases}$$

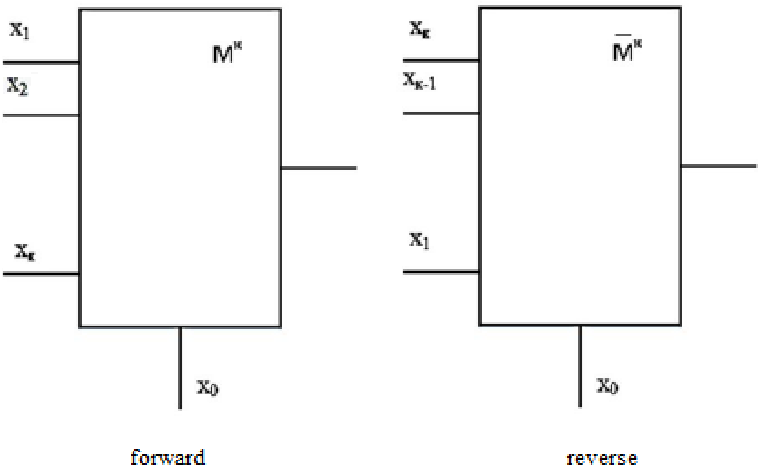

**Fig. 1.** Forward and reverse multiplexers.

A multiplexer with one control signal will be called an elementary multiplexer. Multiplexers with two or more control signals are called complex. In addition to this logical basis, we introduce another logical function - inversion:

$$\overline{X} = \begin{cases} k\text{-}1 \quad if \quad x = 0 \\ k\text{-}2 \quad if \quad x = 1 \\ \text{........................} \\ 0 \quad if \quad x = k\text{-}1 \end{cases}$$

The corresponding logical element is shown in Fig. 2.

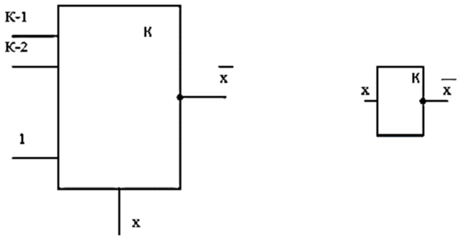

**Fig. 2.** Multi-valued inverter

The functional completeness of the basis can be brought indirectly by obtaining structural diagrams for arbitrary logical functions of the three-valued logic. The table of functioning of the three-valued logical function of one argument is given in Table 1.

**Table 1.** Three-valued logical function of one argument

| x | 0 | 1 | 2 |
|---|---|---|---|
| f(x) | 0 | 2 | 1 |

The rule of functioning of the three-valued logical function of two arguments is presented in Table 2.

**Table 2.** Three-valued logical function of two arguments

| $x_0$ | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 2 |
| $f(x_0,x_1)$ | 2 | 1 | 0 | 0 | 1 | 2 | 1 | 0 | 2 |

Structural diagrams of the three-valued logical function of one and two arguments are shown in Fig. 3, left and right, respectively.

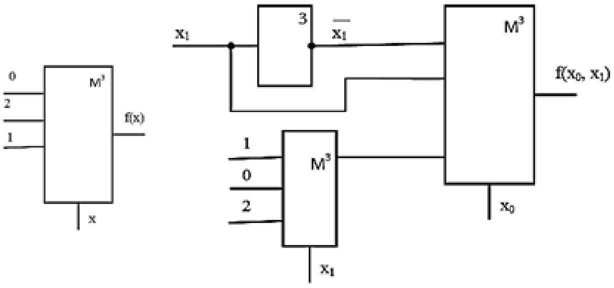

**Fig. 3.** Three-valued logical function of one and two arguments

Similarly, block diagrams using a reverse multiplexer can be obtained. Thus, any logical function of multi-valued logic can be represented by a multi-tiered tree structure in which the basic element is an elementary k-valued multiplexer.

The following equalities hold for any multi-valued logic and can be used to simplify structural diagrams:

$$M_x^k(0,0,...,0) = 0$$

$$M_x^k(1,1,...,1) = 1$$

$$............................$$

$$M_x^k(0,1,2,...,k-1) = x$$

$$M_x^k(k-1,k-2,...,1,0) = \overline{x}$$

Consider an example of the synthesis of the structural diagram of the three-valued adder, the rule of operation of which is given in Table 3 and obtained on the basis of obvious three-valued expressions:

| | | |
|---|---|---|
| 0 + 0 = 0 | 0 + 1 = 1 | 0 + 2 = 2 |
| 1 + 1 = 2 | 1 + 2 = 10 | 2 + 2 = 11 |
| 0 + 1 + 2 = 10 | 1 +1 + 1 = 10 | 0 + 2 + 2 = 11 |
| 1 + 2 + 2 = 12 | 2 + 2 + 2 = 20 | |

**Table 3.** The table of functioning of the three-valued adder

| $b_i$ | 012 | 012 | 012 | 012 | 012 | 012 | 012 | 012 | 012 |
|---|---|---|---|---|---|---|---|---|---|
| $a_i$ | 000 | 111 | 222 | 000 | 111 | 222 | 000 | 111 | 222 |
| $P_{i-1}$ | 000 | 000 | 000 | 111 | 111 | 111 | 222 | 222 | 222 |
| $S_i$ | 012 | 120 | 201 | 120 | 201 | 012 | 201 | 012 | 120 |
| $P_i$ | 000 | 001 | 011 | 001 | 011 | 111 | 011 | 111 | 112 |

where $a_i$, $b_i$ are the symbols that make up i-th digit of the three-valued numbers; $P_{i-1}$ is transfer from the previous digit; $S_i$ is the sum of numbers of the i-th digit; $P_i$ is transfer to the next (senior) digit.

Table 3 allows you to get a block diagram of the full three-valued adder on elementary multiplexers, consisting of a node for realizing the sum of $S_i$ (Fig. 4) and the transfer signal $P_i$ (Fig. 5).

As for the case of binary logic, complex multiplexers with several control signals can be used to construct the structural diagrams of the logical functions of multi-valued logic. It is easy to show that complex multiplexers also have functional completeness.

They can be used to reduce the number of levels in the structural diagram of a device designed to form a model for presenting information on various indicators.

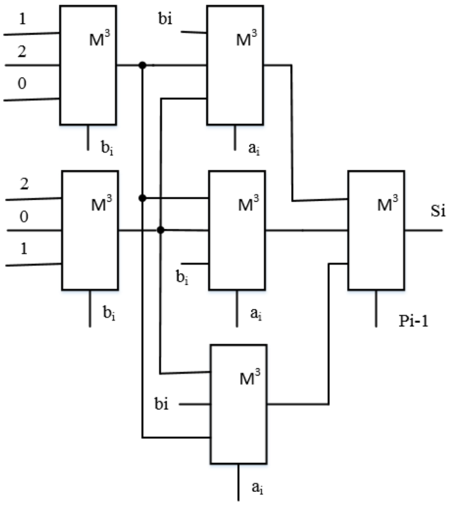

**Fig. 4.** Node for realizing the sum of the full three-valued adder on elementary multiplexers

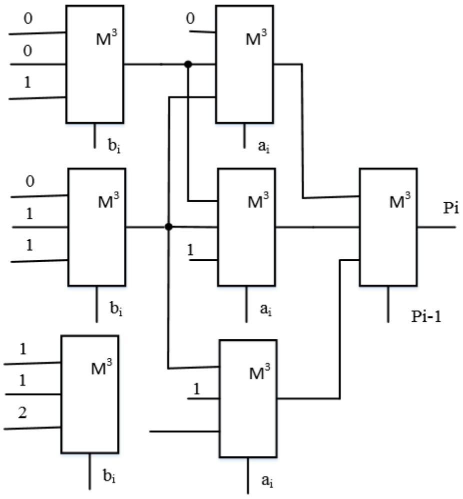

**Fig. 5.** Node for realizing the transfer signal of the full three-valued adder on elementary multiplexers

## 4 Implementation of logic elements with memory in a multiplexer basis

By a memory element is meant an elementary storage element that is capable of storing one bit of a multi-valued code. For k-valued logic, such an element must remember one of the values 0, 1, 2, ..., k-1. Such elements are a trigger circuit of a static or dynamic type. In modern calculators, the static storage mechanism is most often used. A trigger as a memory element must have a certain minimum set of properties that allow it to obtain its functioning table and block diagram [5]. These properties are: storage of the previously set value of the input signal, setting the value 0, setting the value 1, setting the value k-1. For example, a ternary trigger must have the following properties: storage of the previously set value of the input signal, setting value 0, setting value 1, setting value 2. In any multi-valued logic, such properties have a one-bit shift register [5]. The transition table (or graph) is easy to obtain using "window"-technology. For this, the register is conventionally represented as a single “window” through which in the general case a random sequence of k-digit signals alternately passes. The signal value in the "window" determines the state of the register. The number of states is N=k corresponds to the number of values of the used logical function. At each next moment of time, the next signal of the sequence enters the window and replaces (displaces) the previous signal, thereby changing the state of the "window". If the same signal arrives that was in the “window”, the state of the “window” does not change. For three-valued logic, the transition graph has the form shown in Fig. 6 (left - if there is a clock signal; right - in the absence of a clock signal).

A circuit that implements the transition graph will perform the function of a D-trigger. Using the same principle, you can get a k-valued register graph. Given that the transitions of the register triggers occur in the presence of a synchronizing signal C the table of the D-trigger functioning will take the form shown in Table 4.

**Table 4.** The functioning table of the three-valued D-trigger

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Q | 012 | 012 | 012 | 012 | 012 | 012 | 012 | 012 | 012 |
| D | 000 | 111 | 222 | 000 | 111 | 222 | 000 | 111 | 222 |
| C | 000 | 000 | 000 | 111 | 111 | 111 | 222 | 222 | 222 |
| Q | 012 | 012 | 012 | 000 | 111 | 222 | --- | --- | --- |
| ¬Q | 210 | 210 | 210 | 222 | 111 | 000 | --- | --- | --- |

In Table 4, the symbol “-” indicates values that can be arbitrarily selected. According to the Table 4 block diagram of the three-valued D-trigger will look as shown in Fig. 7. All structural schemes are based on a universal logic element, which is a multiplexer. The table shows that all the necessary properties of the trigger as a memory element are satisfied. To build block diagrams in k-valued logic, you can now apply the classical technique developed for binary logic.

As an example, we show the implementation of the structural diagram of the Mealy machine in three-valued logic, which, from the sequence of input signals X = 0, 1, 2, detects a combination of 1220 and sends the output signal y = 1. Otherwise, the machine sends a signal to the output y = 0. The graph of such a finite state machine is shown in Fig. 8, and Table 5 contains a table of its transitions. If the input of the machine, which is in the 3rd state, receives the next input signal equal to 0, then it returns to its original zero state. To implement the structural diagram of this finite state machine, two D-triggers are required, the states of which are denoted by $Q_1$ and $Q_2$. We encode the state of the machine in the form given in Table 6.

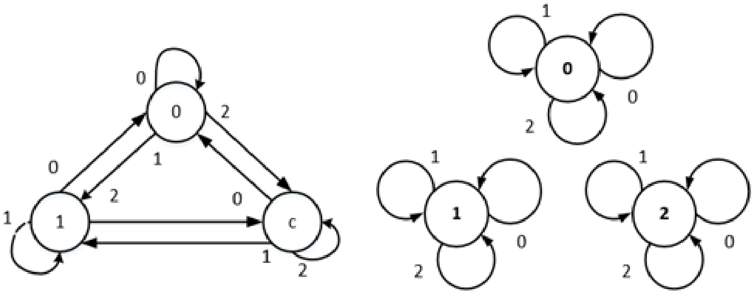

**Fig. 6.** Transition graph of a three-valued shift register

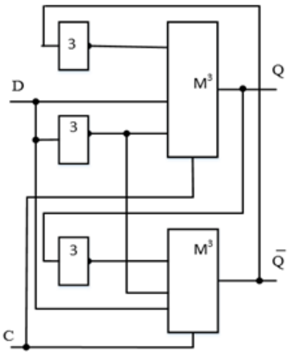

**Fig. 7.** Structural diagram of the three-valued D-trigger

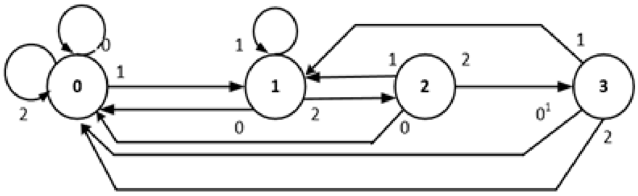

**Fig. 8.** The graph of a state machine that recognizes a combination of 1220

**Table 5.** Transition table of a state machine that recognizes a combination of 1220

| input | state | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|
| 0 | | 0 | 0 | 0 | 0 |
| 1 | | 1 | 1 | 1 | 1 |
| 2 | | 0 | 2 | 3 | 0 |

**Table 6.** Table of the machine state encoding

| state of the machine | | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|
| trigger state | $Q_1$ | 0 | 0 | 0 | 1 |
| | $Q_2$ | 1 | 1 | 1 | 1 |

With the selected option of encoding states, we obtain an encoded table of transitions and outputs presented in Table 7.

**Table 7.** Encoded transition table of a state machine that recognizes a combination of 1220

| $Q'_2$ | 012 | 012 | 012 | 012 | 012 | 012 | 012 | 012 | 012 |
|---|---|---|---|---|---|---|---|---|---|
| $Q'_1$ | 000 | 111 | 222 | 000 | 111 | 222 | 000 | 111 | 222 |
| x | 000 | 000 | 000 | 111 | 111 | 111 | 222 | 222 | 222 |
| $Q_1$ | 000 | 0-- | --- | 000 | 0-- | --- | 001 | 0-- | --- |
| $Q_2$ | 000 | 0-- | --- | 111 | 1-- | --- | 020 | 0-- | --- |
| y | 000 | 0-- | --- | 000 | 0-- | --- | 000 | 1-- | --- |
| $Q_1$ | 000 | 0-- | --- | 000 | 0-- | --- | 001 | 0-- | --- |
| $Q_2$ | 000 | 0-- | --- | 111 | 1-- | --- | 020 | 0-- | --- |

The table corresponds to the structural diagram of a given machine, which consists of three parts: to determine the value of the first digit (Fig. 9), the second (Fig. 10, left) and the third (Fig. 10, right) digit of the state code.

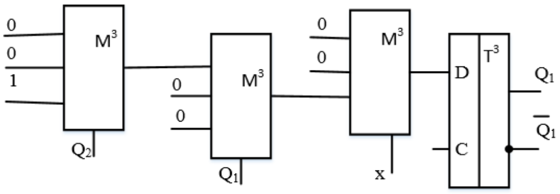

**Fig. 9.** Structural scheme of the machine for determining the value of the first digit

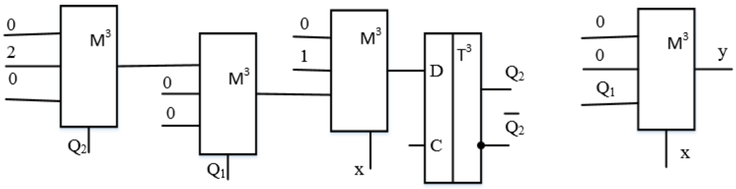

**Fig. 10.** Structural scheme of the machine for determining the values of the second and third digits

## Conclusions

The possibility of using direct and reverse multiplexer circuits as a logical basis for constructing in three-valued logic structural diagrams of digital devices without memory and with memory of any complexity is shown. As an example of using a multiplexer to implement 3-valued logic, a structural synthesis of a finite state machine is given, which, from a sequence of input signals, detects a given combination and generates an output control signal. Using the proposed method for constructing block diagrams of digital devices based on k-valued logic, it will be possible to create control signal conditioners for constructing various models of discrete-analog representation of data on semiconductor indicators. This will increase the capacity, processing speed and display of large amounts of information while reducing hardware and energy costs for their technical implementation.

## References

1. Vladimirova, Yu.: Introduction to Ternary Computer Science. Moscow, Argamak-media (2015).
2. Wakerly, J.: Digital Design: Principles and Practices. 6rd ed. Hoboken, Pearson Education Inc (2018).
3. Knut, D.: The Art of Programming, 3rd edn. vol. 2. Boston, Addison-Wesley (1997).
4. Vladmirova, Yu., Alvarez, R.: On the occasion of the 60th anniversary of the Setun computer. In: X International Scientific Conference named after A. I. Kitov "Information Technologies and Mathematical Methods in Economics and Management" (IT&MM-2019), pp. 144-150. Moscow (2019).
5. Drobik, O., Lobanov, L., Yaskevich, V.: Building digital diagram on multiplexers. Komp'yuterno-intehrovani tekhnolohiyi: osvita, nauka, vyrobnytstvo 8, 16-21. (2012).
6. Ivanko, M., Gasovich, A.: Trinity computers: historical and educational aspects of the study of computer architecture. In: Vestnik MGUP Ivana Fyodorova 1, pp. 42-45. Moscow (2016).
7. Das, S., Sain, J., Dasgupta, P., Sensarma S. Arithmetic Algorithms for Ternary Number System. Procedia Technology 4, 278-285, Elsevier (2012).
8. Lin, S., Kim, Y., Lombardi, B.: CNTFET-Based Design of Ternary Logic Gates and Arithmetic Circuits. IEEE Transactions on Nanotechnology 10(2). 217-225 (2011).
9. Tabrizchi, S., Azimi, N., Navi, K.: A novel ternary half adder and multiplier based on carbon nanotube field effect transistors. Information Technology & Electronic Engineering 18. 423–433 (2017).
10. Mendelson, E.: Introduction to Mathematical Logic. 6th edn. Chapman and Hall (2015).