SYNTHESIS OF LOGIC CIRCUITS WITH PROGRAMMABLE LOGIC CONTROLLER

Antoaneta Hinova
Technical College – Lovech, Bulgaria, hinovi@abv.bg

Abstract: The peculiarities of the synthesis and simulation of logical circuits in the Siemens Programmable logic controller-LOGO are considered. The idea is particularly suitable for studying logical schemes in non-electronic technical specialties due to the specifics of work in the programming environment of this type of controllers and the provided that reduced of the simulation. The main objective is to combine the programming and mastery of the synthesis of logical circuits.
The program environment of the LOGO Kontroler with its eight basic logic functions and 33 special functions allows the study and mastering of the schemes even without the physical use of the controller this is one of the complex controllers, through which are given examples and can be successfully replaced by similar analogues of other companies. Access to the information could be protected by a password. The range includes power models-DC 12 V; AC/DC 24 V and 115/230 V. For some models in the series, the operating voltage limits are very wide, for example LOGO! 12/24RC works with voltages from 10.8 to 28.8 VDC, a LOGO! 230RC with voltages from 85 to 265 VAC. The logical module features are 8 digital inputs, LOGO! 12/24RC and LOGO! 12/24RC Two of the inputs can be configured as analog inputs with a range of 0-10 V. In the event that work with signals 4-20 mA is required, it is necessary to add a precision resistor parallel to the corresponding input. The resistor must be installed in close proximity to the terminals of the module, if possibleThe resistor must be mounted in close proximity to the terminals of the module, if possible. The resistor is 250 cm/0.250 W/0.1%, the input voltage will vary from 1 to 5 V. Other features of the series are: 4 relay outputs (for LOGO! 24 transistor, with built-in short circuit protection; operating temperature range from 0 to + 55 o C; 80 h operation of the internal clock when power is disconnected; 8 logical functions).
Kontroler the Siemens LOGO! Satisfies the requirements for a logical controller for learning purposes on the one hand and the main criteria for selecting a controller on the other. A compact programmable logic controllers used in automation as a relay proxy.
It does not cost more than a few relays and its programming is so flexible as connecting the relay. Its only drawback is the fixed number of inputs and outputs and the small memory for programs and data, but it is especially suitable for getting acquainted with Logical schemes for training in non-electronic technical specialties.
Keywords: Logic circuits, Fusion, programmable logic controller, simulations used in automation as a relay proxy.

1. INTRODUCTION

The proposed programmable logic controller has 8 inputs and 4 relay outputs, so the synthesized circuits are of this magnitude. Each set of input signals corresponds to a set of output signals that do not depend on the previous state of the circuit. In the combination circuits, there can not be a closed loop where a signal from the output of the i-th component, passing through several other logical elements, arrives at the input of the same i-th element. Input signals are converted into intermediate and intermediate - in output, i.e., multistage conversion. Delays in the different steps (steps) lead to a general delay in the -TT scheme. It depends on the speed of the individual logical elements and the number of degrees. The total delay T determines the minimum time during which the signals at the input of the circuit can be changed, ie it determines the discrepancy interval.

Fig.*1. Figure or the LOGO controller*

2. EXPLANATION

After describing a logical device with logical functions it is necessary to compose its corresponding scheme. If there are no specific requirements for the element base, the schematic is constructed using the basic logical function blocks - "And", "Or", "NOT". When evaluating a scheme, the following parameters are considered: number of elements, number of inputs used, number of steps of the scheme. The connection between the inputs and outputs is

Q1 = f1 (I1, I2, ... I8); (1)
Q2 = f2 (I1, I3, ... I8) (2)
Q3 = f1 (I1, I2, ... I8); (3)
Q4 = f4 (I1, I2, ... I8). (4)

The combination scheme has only one binary output, the state of which 0 or 1 depends on the set input variables, for example:

P={not[not(I1+I2)+I2+not I4] +I2}.{not(I1+I2)+I3+not I4+not I2} (5)

Q1=P,

If we lay

P1=not(I1+I2), (7)

P2=not(P1+ I3+I4), (8)

P3=not(P2+I2) (9)

P4=notP2+not I2 (10)

P=P3.P4. (11)

It is possible first to build a common functional pattern of the scheme and then proceed to synthesis (fig.2).

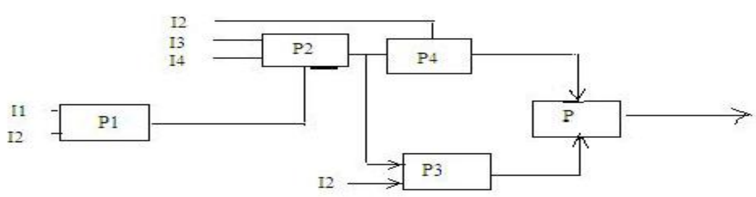

Fig.2. A common functional meta- model of the scheme

Through the basic features of LOGO! the logic function described above can be represented by the exemplary embodiment of fig.3.

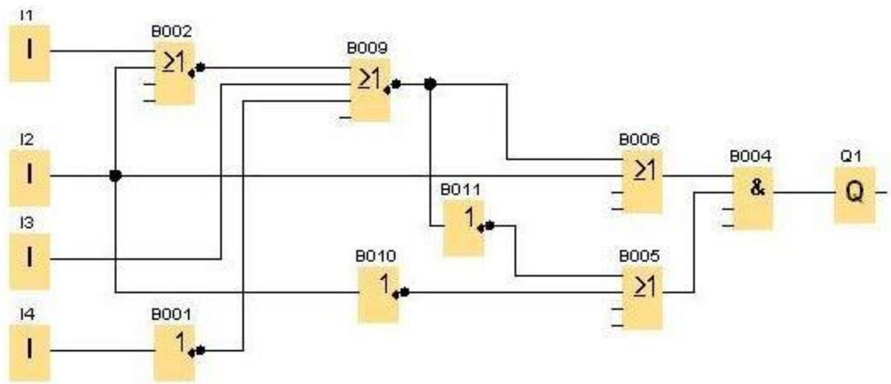

Fig.3. The completed schematic with the basic functional blocks (GF), ready for simulation

The schematic of figure 3 is ready for simulation. The simulation is done virtually in the program environment of the controller, then transferred to execution in the controller itself. If LEDs are placed on the relay outputs of the controller, the action of the synthesized logic circuit (Figure 4) can be observed.

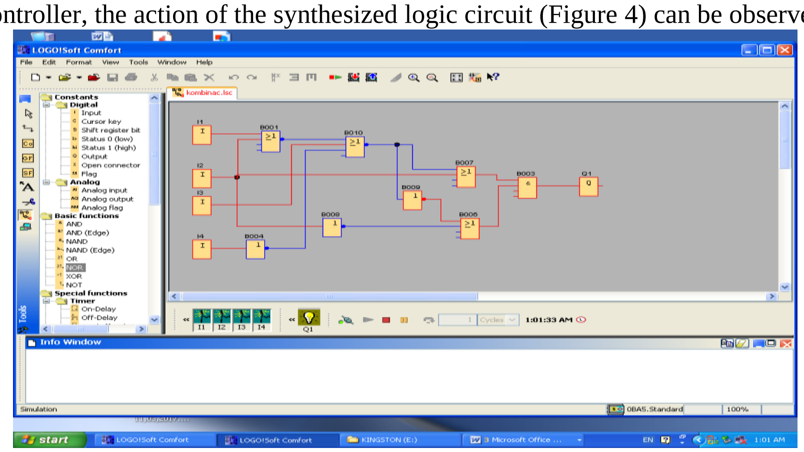

Fig. 4. Simulation of synthesized circuit

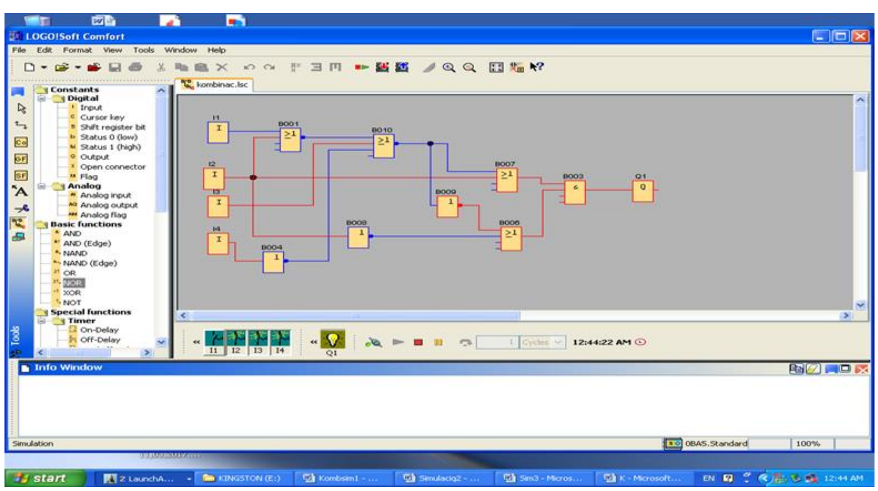

Fig.5 Simulation with three inputs

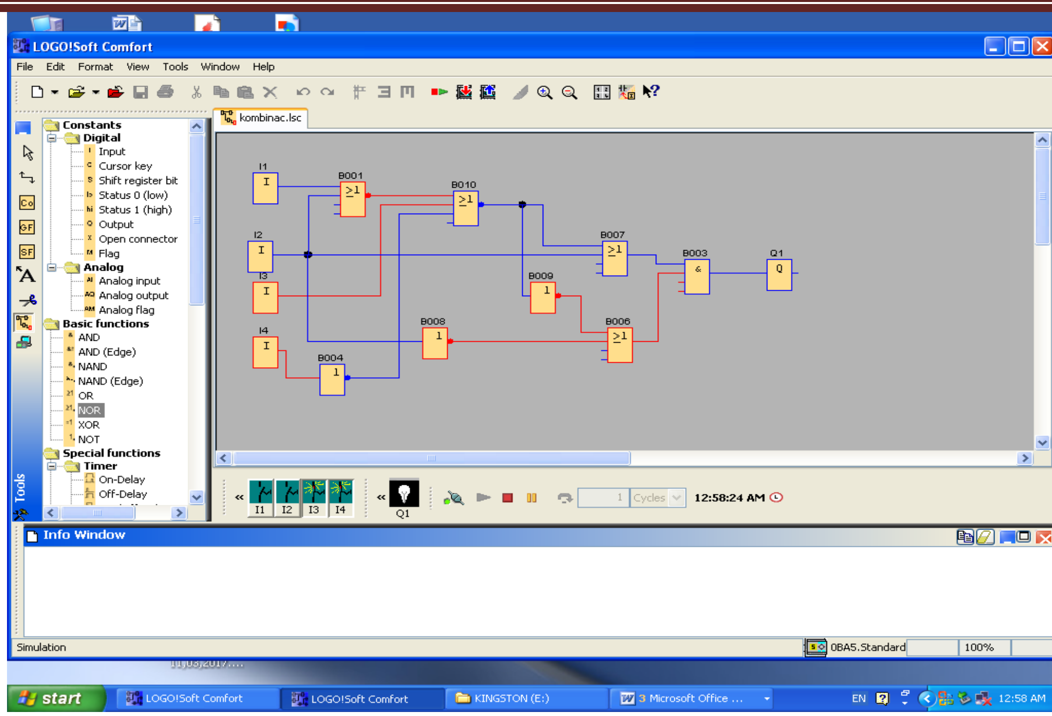

Fig.6 Simulation with two inputs included

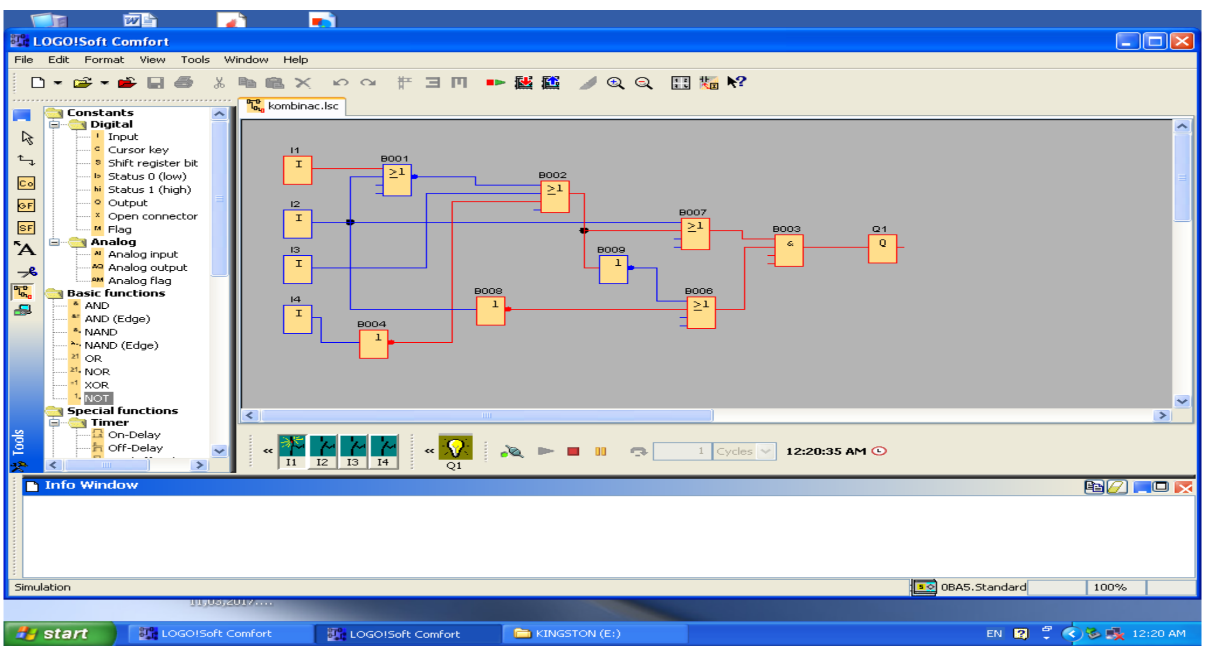

Fig.7 Simulation with one input-I1input

3. Generally, when standard (type) elements are used, different schemes of the same function can be built. They will have a different number of inputs, links, and different complexity. Significant results regarding the optimal synthesis of logic circuits can be achieved by using the minimal coverage methods of logic functions. With regard to the application of the schematic examples, the user is allowed to exhibit creativity in his specific technical field of activity. For example, code converters are combinational schemes that convert an input binary code into a given source code. The number of inputs and outputs of the transducers may be the same or different. The base of code converter synthesis is the truth table that matches the input and output code. If there are unused / forbidden / combinations in the input code, they are used to further simplify the expressions for the source code. If it is necessary to synthesize a code converter from code 8421 into a cyclic code of Gray, it can be set:

$$I1 = Q1 \quad (12)$$

$$Q2 = not\ (I1)\ .I2 + I1.\ not \quad (13)$$

$$Q3 = I2.not\ (I3) + not\ (I2)\ .I3 \quad (14)$$

$$Q4 = I3.not\ (I4) + not\ (I3)\ .I4 \quad (15)$$

The solution with the LOGO programmable logic controller is as follows:

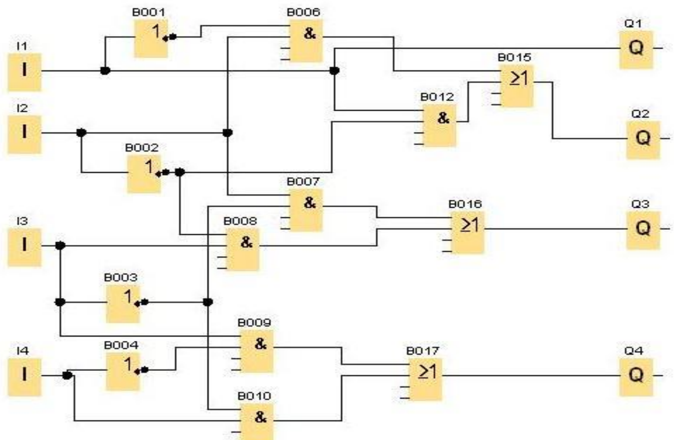

Fig.8.Code converter

4. **CONCLUSION**

The viewer of Siemens LOGO does not run unfinished programs.This is his main advantage. If the simulation is not performed, it is necessary to analyze and adjust the software solution. If part of the scheme works, the simulation shows, that part of the diagram is lit. In lab models, only the final result is usually present. There is no light ndication of individual connections.

When the simulation is performed, the action of the scheme is again analyzed whether it corresponds to the practical task. This is also aided by the built-in weekly clock function block. In the absence of a controller, only its program portion can be loaded. Button simulation, which closes the normally open contacts and the lights on the outputs, is very useful to illustrate the action of the program. The simplicity of servicing the LOGO module through a series of simple practical tasks to master one of its programming modes and, depending on the purpose, to understand in practice the operation of the basic logic functions.

LITERATURE

Floyd Thomas L.(2008*)*, *Digital Fundamentals*, ISBN 0-675-21217-0, Macmilian Publishing Company, New York
Hinova A.(2015). *Electronic Models for Supporting Technical and Vocational Education*, ISBN 978-619-160-431-9, Avangard Prima, S., pp. 25-28
Hinova A. I.(2013) *Electronic devices and models for practical training of non-electronic technical specialties*, Dissertation, Technical University of Sofia - IPF - Sliven
Hristov V., M. Gilevski,(2018) *Expanding the Programming Options for Programmable Logic Devices through*
Ilarionov R. (2007)*Digital Circuits*, ISBN 978-954-9770-98-8, Gabrovo print EOOD
Matlab, Yearbook of the Technical University - Sofia, vol. 68, no. 2, 79-88, Sofia, ISSN 13110829.