Edward HRYNKIEWICZ, Mirosław CHMIEL

Silesian University of Technology, Institute of Electronics

Programmable Logic Controller - Basic Structure and Idea of Programming

Abstract. *The paper describes a structure of PLC and explains an idea of PLC programming on the base of simple example. The authors start from micro-programmed digital circuit designed for belt conveyor control and next equipping it with set of function modules as timers, counters, AD and DA converters and microprocessor finish their description at the bit-byte PLC structure. The utilization of function modules is shown on the base of timer application in control program of set of the conveyors.*

Streszczenie. *W artykule została przedstawiona struktura sterownika programowalnego i wyjaśniona na prostym przykładzie idea programowania takiego sterownika. Prezentację struktury sterownika autorzy rozpoczynają od prostego mikroprogramowanego układu zaprojektowanego do sterowania przenośnikiem taśmowym. Następnie rozbudowując układ o takie moduły funkcyjne jak timery, liczniki, przetworniki A/C i C/A oraz mikroprocesor dochodzą do struktury sterownika typu bit-bajt. Wykorzystanie modułów funkcyjnych zostało przedstawione na przykładzie timer'a w programie sterowania zestawu przenośników. (**Programowalny sterownik logiczny – podstawowa struktura oraz idea programowania**).*

Keywords: micro-programmed digital circuit, PLC structure, PLC programming.
Słowa kluczowe: mikroprogramowany, struktura programowanego sterownika logicznego (PLC), programowanie sterowników PLC.

Introduction

Programmable logic controllers (PLCs) are present in industry since sixties years of the previous century. Its are used for control different processes in steel plants, mains, in automotive, mechanical and chemical enterprises and so on. A lot of PLCs one can meet in intelligent buildings too. Due to this fact they should be universal. An universality is obtained by re-configurability (the PLCs are equipped with different I/O modules) and programmability (the PLCs are specialized for given control task by configuring and programming). It seems that the programmable logic controllers are well known but their structure and the way of operation are still the subject of a lot of questions. They are described in many publications [1, 2, 3, 4, 5, 6, 7] but typically not in details.

To present a structure of typical PLC and to explain an idea of PLC programming first let us consider the following example.

Example 1

Let us design a control unit for belt conveyor engine. The conveyor control unit contains engine overheating switch off (OH), emergency switch off (ER), normal switch off button (R) and switch on button (S). The state of conveyor operation is controlled by CR relay contacts and signaled by the lamp (L). The ladder diagram of the unit can be seen in fig.1. The operation of the control unit can be described by means of state diagram presented in fig.2.

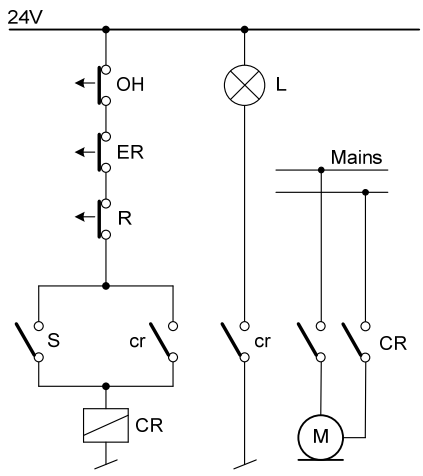

Fig.1. The ladder diagram of conveyor control unit

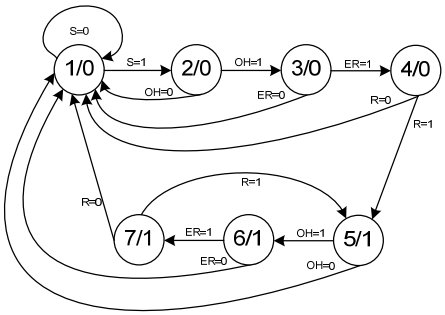

Fig.2. State diagram for example 1

State encoding for state diagram from fig.2 can be written as follows:

1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111

Table 1. The micro-program representing the state diagram from fig.2

State	Actual address				Data				
	Q_0	Q_1	Q_0	Cond.	Next address			CR	L
	A_3	A_2	A_1	A_0	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1	0	0
	0	0	1	1	0	1	0	0	0
2	0	1	0	0	0	0	1	0	0
	0	1	0	1	0	1	1	0	0
3	0	1	1	0	0	0	1	0	0
	0	1	1	1	1	0	0	0	0
4	1	0	0	0	0	0	1	0	0
	1	0	0	1	1	0	1	0	0
5	1	0	1	0	0	0	1	1	1
	1	0	1	1	1	1	0	1	1
6	1	1	0	0	0	0	1	1	1
	1	1	0	1	1	1	1	1	1
7	1	1	1	0	0	0	1	1	1
	1	1	1	1	1	0	1	1	1

The state diagram presented can be implemented in the micro-programmed circuit seen in fig.3, while the micro-program stored in ROM is written in table 1 [9, 10].

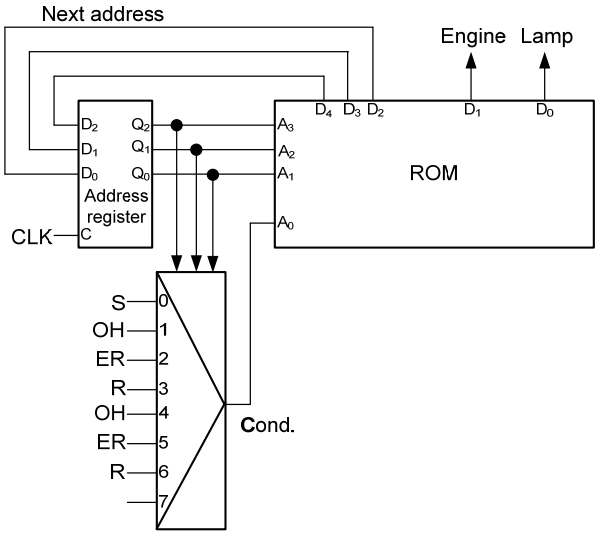

Fig.3. The micro-programmed conveyor control unit

The structure of a simple PLC

In solution shown in fig.3. a valid condition is chosen by means of an address of current state. More universal addressing a valid condition may be done directly from the program memory. It means that a condition test instruction is stored in a memory together with the condition address. The states of output signals may be set or reset in the same way. To create possibility for testing the output states and for designing sequential automata the outputs should be connected to the condition multiplexer too, as it was shown in fig.4. Moreover the program counter (PC) was applied instead of address register to avoid of storing an address of a next state in the program memory. On the other hand to create possibility of jump execution in control program the jump instruction was introduced that is stored in control program memory together with new state of program counter. The condition flip flop (CFF) is set at the beginning of series of test instructions. Such arrangement supports calculation of logic product (conjunction) of binary variables - inputs and outputs for example. This way we obtain simple programmable logic controller (PLC) which may be called logic or bit processor [10, 11].

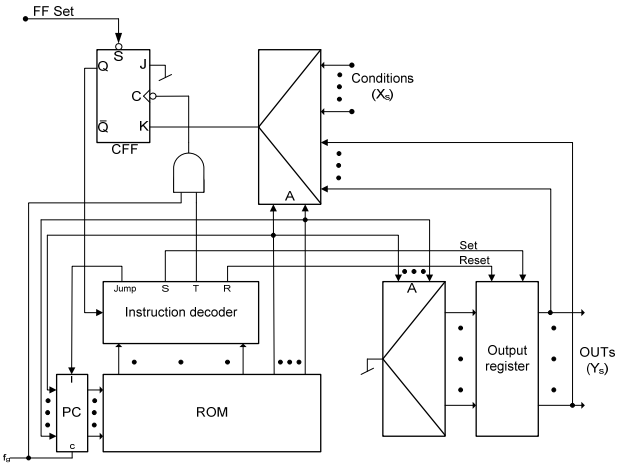

Fig.4. The structure of simple programmable logic controller

A PLC with counter and timer

Example 2

At the moment let us consider more complicated situation created by set of two conveyors - one after an other (see fig.5). To avoid overloading of transformer station (because of transient states) the first conveyor has to be started for example 10 [s] after the second one. It means that a control unit should be equipped with timer which can measure-off proper time interval. Moreover it sometime happens that it is necessary to count some events which occur in a controlled process. To provide such functionality a control unit should contain a counter module. The connection timer and counter modules in the control unit structure is shown in fig.6.

In a system the instructions are represented by the code words. The exemplary instructions coding is shown below:

```
Test              -   0001
Set               -   0010
Set(CFF=1)        -   0011
Set(CFF=0)        -   0100
Reset             -   0101
Reset(CFF=1)      -   0110
Reset(CFF=0)      -   0111
Jump              -   1000
Jump(CFF=1)       -   1001
Jump(CFF=0)       -   1010
NOP               -   0000
```

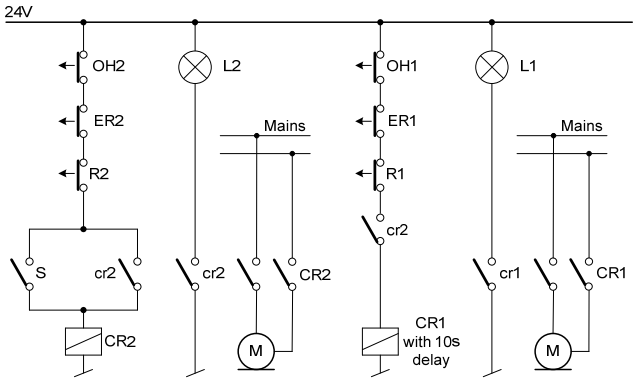

Fig.5. The ladder diagram of two conveyors control unit

Using the assumed instruction coding and exemplary addressing of inputs and outputs one can write the control program for set of two belt conveyors in the following form:

```
                                 line    instr.
                                 number  code    addr.
0  Test         OH2              00000   0001    00001
1  Jump(CFF=0)  line15           00001   1010    01111
2  Test         ER2              00010   1010    01111
3  Jump(CFF=0)  line15           00011   1010    01111
4  Test         R2               00100   0001    01010
5  Jump(CFF=0)  line15           00101   1010    00011
6  Test         S                00111   0001    01111
7  Set(CFF=1)   CR2              00111   0011    01101
8  Set(CFF=1)   L2               01000   0011    01110
9  Set(CFF=1)   Timer            01001   0011    00111
10 Test         Timer            01010   0001    00111
11 Jump(CFF=0)  line0            01011   1010    00000
12 Set          CR1              01100   0010    00101
13 Set          L1               01101   0010    00110
14 Jump         line0            01110   1000    00000
15 Reset        CR2              01111   0101    01101
16 Reset        L2               10000   0101    01110
17 Reset        CR1              10001   0101    00101
18 Reset        L1               10010   0101    00110
19 Reset        Timer            10011   0101    00111
20 Jump         line0            10100   1000    00000
```

It can be noticed that at line 20 it is jump to line 0. It means that control program is executed in serial-cyclic way. Such way of control program execution is typical for programmable logic controllers.

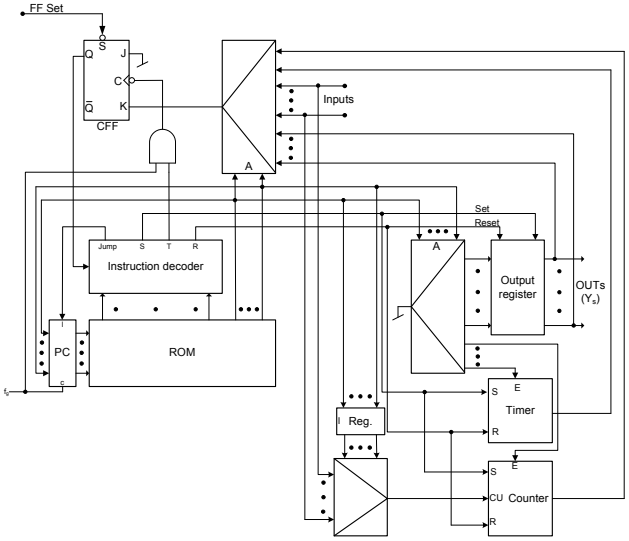

Fig.6. The PLC with a timer and counter

A developed structure of PLC

The programmable logic controller presented above is able to control binary objects. To control continuous (analogue) objects the unit should be equipped with AD and DA converters and at least Arithmetic Logic Unit (ALU) converters for processing numeric values. In more sophisticated constructions a microprocessor/ microcontroller may be used for byte/word operation instead of arithmetic logic unit [12]. Such PLC structure is called bit-byte [13, 14] because bit operations are executed by above described fast bit-processor while byte/word operations are executed in software way by microprocessor The general block diagram of such PLC is shown in Fig.7.

Conclusions

In the paper the authors describes a structure of PLC and explains an idea of PLC programming on the base of simple example. They start from simple micro-programmed digital circuit designed for belt conveyor control and next equipping it with set of function modules as timers, counters, AD and DA converters and microprocessor finish their description at the bit-byte PLC structure The utilisation of function modules is shown on the base of timer application in control program of set of the conveyors.

The processors in bit-byte structure of PLC can execute their operations concurrently. Very interesting problem is how to put both processors in concurrent operation as far as it is possible? More about this problem a reader can find in [15, 16]. It is necessary to added that a PLC programming was standardized. The reader should refer to the standard EN/PN 61131-3 where control program representation forms are described [17].

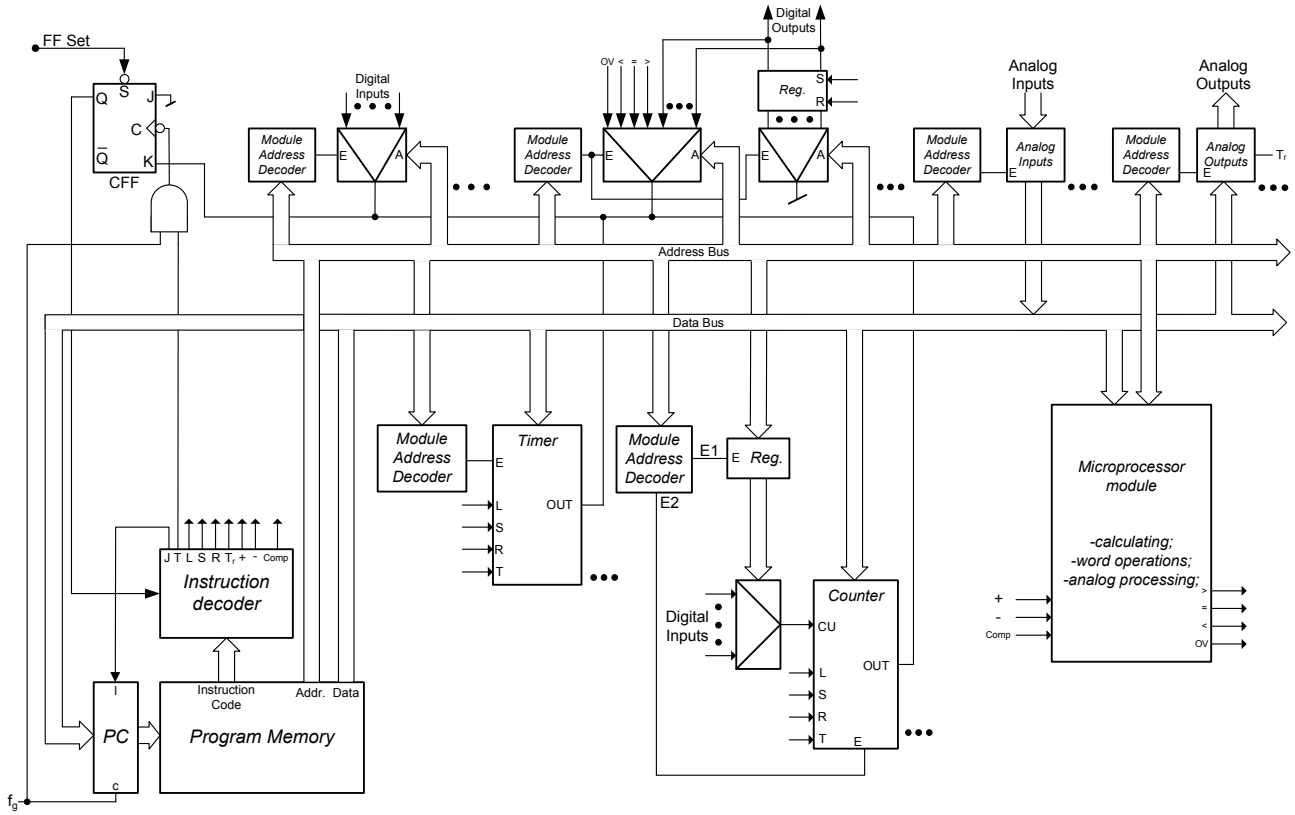

Fig.7. The developed structure of the PLC

The works was supported by Ministry of Science and Higher Education of Republic of Poland No, 5391/B/T02/2010/38

REFERENCES

[1] Broel-Plater B., Sterowniki programowalne, właściwości i zasady stosowania (Programmable Logic Controllers, Features and Rules of Application). Seria Tempus. Szczecin, Wydział Elektryczny Politechniki Szczecińskiej, Poland, (2000)
[2] Petruzella F.D., Programmable Logic Controllers, Fourth Edition. Publisher: Mc Graw-Hill, USA, (2011)
[3] Irwin J.D., Industrial Electronics. Handbook. A CRC Handbook Published in Cooperation with IEEE Press (1997)
[4] Bolton W., Programmable Logic Controllers, Fifth Edition. Elsevier Ltd, (2009)
[5] Warnock I.G., Programmable Controllers. Operation and Application. Hemel Hempdtead, Prentice Hall Int., (1998)
[6] Legierski T., Kasprzyk J., Wyrwał J., Hajda J., Programowanie sterowników PLC (Programming PLC). Wydawnictwo Komputerowe Jacka Skalmierskiego, Gliwice, (2008)
[7] Chmiel M., Mocha J., Hrynkiewicz E., Milk A., Central Processing Units for PLC implementation in Virtex-4 FPGA. *Proceedings of the 18th IFAC World Congress,* August 28-September 2, Milano, Italy, (2011)
[8] Łuba T., Układy mikroprogramowane (Microprogrammed circuits). Oficyna Wydawnicza Politechniki Warszawskiej, Poland, (1996)
[9] Molski M., Modułowe i mikroprogramowalne układy cyfrowe (Module and microprogrammed digital curcuits). WKŁ, Warszawa, Poland, (1986)
[10] Donandt J., Improving response time of Programmable Logic Controllers by use of a Boolean coprocessor, *IEEE Comput. Soc. Press.* 4:167-169, Washington, DC, USA, (1989)
[11] Aramaki N., Shimokawa Y., Kuno S., Saitoh T., Hashimoto H., A new Architecture for High-performance Programmable Logic Controller, *Proc. of the IECON'97 23rd Inter. Conf. on Industrial Electronics, Control and Instrumentation*, IEEE part vol.1, 187-190, New York, USA, (1997)
[12] Milik A., Hrynkiewicz E., PID Module for Reconfigurable Logic Controller. *Preprints of the IFAC Workshop on Programmable Devices and Systems*, PDeS 2000, Ostrava, Czech Republic, February (2000)
[13] Chmiel M., Hrynkiewicz E., Remarks on Parallel Bit-Byte CPU structures of Programmable Logic Controllers. In: Design of Embedded Control Systems, Section V, (Adamski M.A., Karatkevich A., Węgrzyn M), 231-242, Springer Science + Business Media, Inc., (2005)
[14] Chmiel M., Mocha J., Hrynkiewicz E., A FPGA-Based Bit-Word PLC CPUs Development Platform, *The International IFAC Workshop on Programmable Devices and Embedded Systems,* PDeS'10, October 6-7, Pszczyna, Poland, 155-160, (2010)
[15] Chmiel M., Hrynkiewicz E., Milik A., Concurrent operation of the processors in Bit-Byte CPU of a PLC, *Preprints of the IFAC World Congress*, Prague, Czech Republic, July 3-8, (2005)
[16] Chmiel M., Hrynkiewicz E., Fast Operating Bit-Byte PLC, *Preprints of the 17th IFAC World Congress* (on DVD-ROM), Seoul, Korea, July 6-11, 14810-14815, (2008)
[17] Norm EN 61131-3:2004(U). Programmable Controllers – Part 3; Programming Languages, I*nternational Electrotechnical Commission*, Geneva, (2004)

***Authors**: dr hab. inż. Edward Hrynkiewicz, profesor Politechniki Śląskiej, Instytut Elektroniki, ul. Akademicka 16, 44-100 Gliwice, E-mail: Edward.Hrynkiewicz@polsl.pl;*
dr inż. Mirosław Chmiel, Politechnika Śląska, Instytut Elektroniki, ul. Akademicka 16, 44-100 Gliwice,
E-mail: Miroslaw.Chmiel@polsl.pl.