

Development of an n -I/O PLC Ladder Diagram Translator

Engr. Conrad A. Populi¹, Engr. Shearyl Arenas¹

¹College of Engineering and Architecture, Technological Institute of the Philippines
(E-mail: cpopuli.ece@tip.edu.ph, shearyl.arenas@tip.edu.ph)

Abstract—The Programmable Logic Controller or PLC has been used in industrial automation for decades, and the demand for these has continued to rise through the years. A simple PLC system can be easily programmed using a ladder diagram. However, when designing complex systems involving several I/O devices, engineers and electricians are faced with difficulties and frustrations. Normally, they use Karnaugh Map to simplify their designs. And in recent years, Quine McCluskey method is gaining popularity among engineers. But these methods are tedious, time-consuming and prone to error when done manually. Although tools already exist, the supported number of inputs and outputs is limited. And their output is only the simplified Boolean Expression. This paper will describe a software tool that would accept more than 10 input and output values that will be simplified using Quine McCluskey method. And the result will be translated into a ladder diagram.

Keywords—PLC; Ladder Diagram; Quine McCluskey method

I. INTRODUCTION

Programmable Logic Controller or PLC is an industrial control system used mainly in production processes, wherein it monitors and controls an extensive system of I/O devices. Although there are other ways to program a PLC that is standardized in IEC 61131-3, ladder diagram is the most common and widely accepted form. It is a graphical programming language that represents devices in the relay logic in a “ladder”-like diagram.

The design can be straightforward with a few input and outputs. However, this can become complicated as the functions that the PLC performs increases. So it is common for engineers and technicians to simplify their design several methods. The most common of these methods is the Karnaugh Map. The Karnaugh Map or K-Map can be described as a two-dimensional truth table, wherein the variables are easily grouped together to eliminate redundancy. It is practical for up to 5 or 6 variables but not for complex designs with more variables. For these kind of systems, alternative methods, such as the Quine McCluskey method is commonly used. Quine McCluskey is a tabular method of simplification by reducing the derived set of prime implicants of the function.

Since the Quine McCluskey algorithm is easier to implement in a computer program as compared to K-Map, there are several tools available online. Although these are very useful for beginners since the presentation of the process is

easy to understand, it is not so practical for a more complex design because of the limited number of variables supported. Furthermore, the results are mostly presented in its Boolean Expression form, and no ladder diagrams created. There is one notable online tool, the “Logic Circuit Simplification” from www.32x8.com. In this website, the result is presented in a digital logic circuit. However, as for the supported inputs, a maximum of 8 inputs are allowed for the Karnaugh Map, and only up to 6 inputs for the Quine McCluskey method.

II. THE QM LAD SYSTEM

The QM LAD System (QLS) is a simple offline application tool created in Visual Basic .NET. This accepts any number of input and output, simplifies the function using the Quine-McCluskey Algorithm, and presents the output in ladder diagrams and boolean expressions.

A. Main Input Screen

The QLS starts with the Main Input Screen, as shown in Figure 1. The screen is divided into three sections – the Menu Section, Project and System Information Section and Truth Table Input Section.

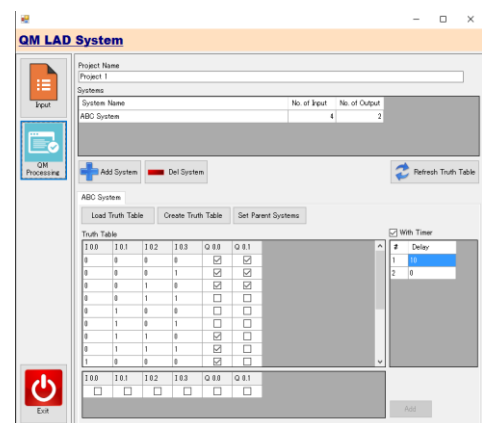


Figure 1. Main Input Screen

1) *Menu Section.* This section contains the different menu buttons that are used in QLS. The first button, which is the “Input” menu, clears the input in the Project and System Information Section. Upon pressing this button, a confirmation message will be displayed and will ask whether the user wants to erase all the data input. This is to ensure that the user did not press it by mistake.

The second button, which is the “QM Processing” menu, will prompt the start of the processing of the data entered in the Truth Table Input Section. Upon pressing this button, the data

inputted will be checked for any errors. If no error was found, the input will be simplified using the Quine McCluskey Algorithm, and the resulting ladder diagram will be displayed at a separate window.

Finally, the last button is the “Exit” button. When pressed, it closes all opened screens and terminates the software.

2) *Project and System Information Section.* In this section, the user can set the project and system information. In the “Project Name” field, the user can set the name of the Project. This will serve as the Title of the succeeding screens of the software. As for the settings of the systems, the user can change the number of systems that a project has by clicking the “Add System” and “Del System” buttons. For each system, the user can set its name and its corresponding number of inputs and outputs.

3) *Truth Table Input Section.* Initially, this section does not display anything. After the user enters the data in the previous section (Project and System Information Section), the Truth Table Input section is refreshed upon pressing the “Refresh Truth Table” button. Depending on the selected input-mode that is selected in the screen, the Truth Table Input section may or may not display any data.

There are two input modes that a user can specify the values of the input and output – the auto-create of truth table values, and the manual input.

The first is by loading the entire truth table through the “Load Truth Table” button. However, this function is only available for those systems that have at most 10 inputs. Because of the amount of values of those with more than 10 inputs, the process of adding these into the truth table takes a lot of time; thus, affecting the performance of the software. As the number of input increases, the amount of data to be loaded doubles. For instance, a 10-input system has 2^{10} or 1,024 rows of data. But for an 11-input system, it is already 2^{11} or 2,048 rows of data. The difference between the two systems is just 1 input variable, but the amount of data doubled. Furthermore, the user would also have a hard time setting the conditions considering the amount of data in the truth table.

On the other hand, the second method is through manual input of values. The user specifies which values are “on” or “off”. Upon clicking the “Add” button located at the side of the table, the values are inserted in the truth table. If the input of an output is dependent on other systems, this could be established in the “Parent Systems Setting” screen.

B. QM Simplifier Process

The QLS uses the Quine McCluskey algorithm in simplifying the entered input values. Since QLS supports more than 5 variables, Karnaugh Map cannot be used in the simplification process.

The flow of the process is described in the following steps.

- 1) Find all the prime implicants from the input.
 - a) Get all the input values where output value is “1”. These are the minterms.

- b) Group the minterms that has equivalent number of 1’s. For example, 0011 is grouped with 1010 and 1100, while 1011 is grouped with 0111.

- c) Compare the values of each group, starting from the two groups with the least value of 1s. If the two numbers are the same except for one digit, these two digits are considered paired. Merge these numbers by copying the same digits and replacing the digit that was different with a “-“. Place the new number into a new list of implicants.

- d) Repeat step c until the implicants cannot be merged anymore.

- 2) Find the minimum set of prime implicants to cover all the input by plotting minterms against the implicants.

- a) Identify the essential prime implicants. These are the prime implicants that covers only one minterm.

- b) After identifying the essential prime implicants, get the minimum set of prime implicants that would cover the remaining minterms.

C. Result Screens

The result will be presented in two forms – in a ladder diagram and in its Boolean Expression form.

1) *Ladder Diagram Screen.* After the processing of the Quine McCluskey method, the result is interpreted and the ladder diagram is plotted, as shown in Figure 2. For this research, the symbols and labels of each component is according to the standard used for Siemens S7-200. Thus, the input is represented by the identifier I, while the output is represented by the identifier Q.

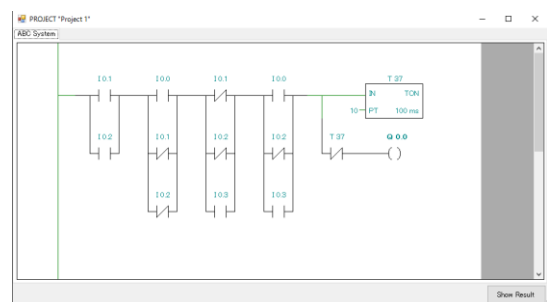


Figure 2. Ladder Diagram Screen

For each output of each systems, the Ladder Diagram Translator modules interprets the Boolean expression and draws the corresponding symbols in this screen. The symbols are mapped according to their possible position in the ladder. Also, since dependencies are possible in QLS, the Translator module also illustrates the labels of each dependencies, as well as the necessary timer symbols.

2) *Boolean Expression Screen.* Aside from displaying the Ladder Diagram of the result of the QM Algorithm, the user may also see the result in its Boolean Expression form, as illustrated in Figure 3.

Systems	Q 0,0	Q 0,1
ABC System	$BC + AB'C' + B'C'D + AC'D$	$BC + ABD + B'C'D$

Figure 3. Boolean Expression Screen

III. CONCLUSION AND FUTURE WORKS

With the continuous increase in the demands for automation in modern industry, efficiency in the design and implementation could save valuable resources for companies.

The QM LAD System is a useful tool that could aid technicians and engineers in their design work. The difference with the some existing tools is that QLS directly simplify the design and outputs the corresponding ladder diagram. It supports more than 5 variables for inputs and outputs, which is valuable for complex designs. And since it also includes automatic generation of ladder diagrams, the amount of time that engineers and technicians spend in simplification and design could improve significantly. Furthermore, with the simplicity of the design of the screens and ease of use of this application tool, it could also be utilized for educational and training purposes.

Subsequent improvements can include support for “Don’t cares” in the input, and export and printing capabilities for the ladder diagram. Also, since the Ladder Diagram Translator uses mapping to correctly illustrate the symbols, other symbols and labels of other PLC devices can be added. Further research and development may eventually include an export functionality to directly program a PLC.

REFERENCES

- [1] S. Banerji, “Computer Simulation Codes for the Quine McCluskey Method of Logic Minimization,” arXiv preprint arXiv:1404.3349.
- [2] H. R. Hwa, “A method for generating prime implicants of a Boolean expression,” IEEE Transactions on Computers 23.6, 1974, pp. 637-641.
- [3] T.K. Jain, D.S. Kushwaha, and A.K. Misra, “Optimization of the Quine-McCluskey Method for the Minimization of the Boolean Expressions,” in Fourth International Conference on Autonomic and Autonomous Systems (ICAS 2008), 2008, pp. 165-168.
- [4] S.P.Tomaszewski, I.U. Celik, and G.E. Antoniou, “WWW-based Boolean Function Minimization,” Int. J. Appl. Math. Comput. Sci., vol. 13, no.4, pp. 577-583, 2003.
- [5] J. Vitthal, and A. Buchade, “Modified Quine-McCluskey Method,” arXivpreprint arXiv:1203.2289.



Shearyl U. Arenas received the B.S. in electronics and communications engineering from Technological Institute of the Philippines, (T.I.P.) Quezon City, Philippines in 1997, and the master of engineering from Technological University of the Philippines, Manila, Philippines in 2003. She is currently pursuing the Ph.D. in technology management at Technological University of the Philippines, Manila. Since 1998, she has been with the Electronics Engineering department of T.I.P., where she is currently a Program Chair. She is a member of IEEE and Institute of Electronics Engineers of the Philippines. She was conferred as ASEAN Engineer by ASEAN Federation of Engineering Organizations (AFEO) in 2014.



Conrad A. Populi is pursuing his Masters degree in Electronics Engineering at Technological Institute of the Philippines in Quezon City (Philippines). He worked as an Instrumentation Commissioning Technician in Saudi Arabia, Qatar and Abu Dhabi for 2 years. He is currently working as the OJT Faculty Coordinator for the undergraduate students in TIP.