A Study on a Hardware Translation Tool for PLC Programs

Graduate Advisor Shuichi Ichikawa 013703 Ryou Ikeda

1 Introduction

Programmable Logic Controller (PLC) is a kind of computer, which is used for sequence control of manufacturing systems. Though PLC is flexible, the performance of PLC is not always sufficient for large systems.

Field Programmable Gate Array (FPGA) is a reconfigurable logic LSI. Implementing control logic with a FPGA to replace a PLC, response time can be improved while sustaining the flexibility of control.

Miyazawa et al.[1] proposed a method to translate PLC programs into VHDL programs. This method only deals with simple logic functions such as AND, OR, NOT, and flipflop. They examined small and simple examples, but gave no close consideration to actual PLCs and applications.

In this study, actual sample programs of FX2N PLC are examined to realize a practical translation system. A framework is presented to implement a whole control system with FPGA technology. The performance improvement is also shown with sample control systems.

2 Translation of PLC program to hardware description

Ladder diagrams are widely used to describe PLC programs. Figure 1 shows an example of ladder diagram, where output coil Y000 is enabled when both switches X000 and X002 are turned on. A ladder diagram consists of *stages*. Each stage is composed of a condition part and a process part as shown in Figure 1. In process part, various PLC components and instructions can be described instead of output coils.

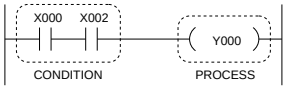
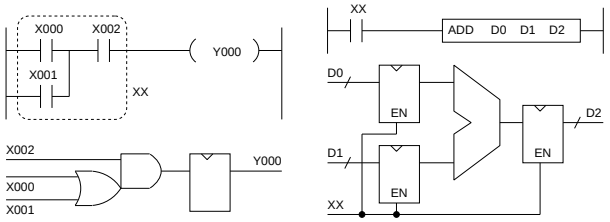


Figure 1: A ladder diagram

In this study, hardware implementation of bit operations, data transfers, edge detections, arithmetic operations, and timers are examined. As the operands of these instructions, switch X, coil Y, internal relay M, data register D, constant K, timer T can be handled. Figure 2 shows two examples of hardware translation of ladder diagram.



(a) Bit operation (b) Arithmetic op.

Figure 2: Ladder diagram vs. hardware

In Figure 2(b), the operands of ADD instruction are connected to input registers, which are fed to an adder. The condition part is connected to enable ports of flipflops.

The stages of PLC program are executed sequentially from top to bottom. Thus, it is straightforward to implement a sequential circuit, each state of which corresponds to each stage of a ladder diagram.

3 Implementation of tools

Figure 3 illustrates the framework of tools, which translate, integrate, and implement the logic circuits of control system onto a FPGA. In Fig. 3, double rectangles designate three tools implemented by the author. The arrows of solid line designate the path to generate logic circuits from control logic, while the arrows of broken line designate the path for performance evaluation.

The framework was verified by using motor control logic as an example. First, its control logic is designed with Mitsubishi GX Works software (step A in Fig. 3), and it is translated into

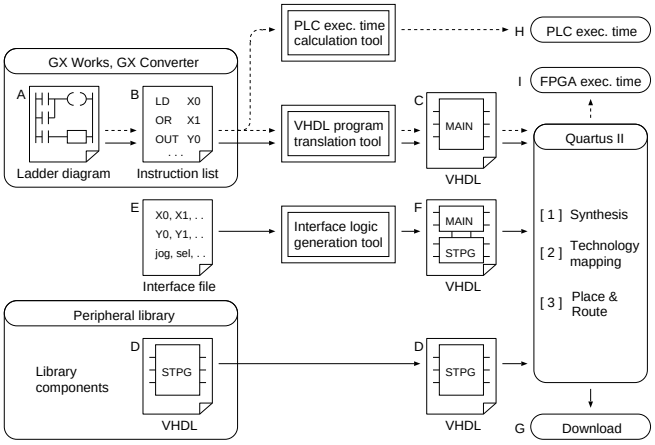


Figure 3: Framework of tool set

instruction sequence with GX Converter (B). Then, hardware translation tool translates the instruction sequence into the hardware design described in VHDL (C). Users might write the top-level design (F) by themselves, or they prepare the interface description file (E) to generate the top-level design by using interface logic generation tool. Finally, Altera Quartus II 4.0 software processes a top-level design(F), a control logic (C), and a pulse generator STPG (D) to generate a bitstream, which is downloaded onto a target FPGA (G).

4 Evaluation

PLC systems and FPGA systems were examined and evaluated for two example control systems (A and Y). The evaluation results are summarized in Table 1.

The evaluation flow is shown as broken lines in Figure 3. First, PLC execution time calculation tool estimates the scan time (H) of a control program, which is derived from PLC instruction sequence (B). Second, PLC instruction sequence is translated into the corresponding hardware design (C) by using VHDL translation tool. Then, Quartus II software generates the FPGA design for Altera APEX20KE FPGA, with an estimated operational frequency. Execution time on FPGA (I) is estimated as the product of clock cycle time and the number of states.

Table 1: The result of evaluation using sample

Sample		Exec. time (sec)	Area	Steps
A	PLC	1.25×10^{-3}	—	538
	FPGA	4.36×10^{-5}	4242 LEs	—
Y	PLC	5.89×10^{-4}	—	93
	FPGA	1.21×10^{-7}	68 LEs	—

5 Conclusion

Derived hardware designs well fit in a APEX20KE FPGA device, while the performance was much improved by using FPGA. The following items are left for future works: (1) more samples should be examined, and (2) the translation tool should be enhanced to support more functions and to derive more performance.

References

[1] I. Miyazawa et al.: "Implementation of Ladder Diagram for Programmable Controller Using FPGA," Proc. IEEE Int'l Conf. Emerging Technologies and Factory Automation, Vol. 2, pp. 1381–1385 (1999).

[2] Mitsubishi Electric Corp.:"FX1S/FX1N/FX2N/FX1NC/FX2NC series programming manual," http://wwwf3.mitsubishielectric.co.jp/members/o_manual/plc_fx/jy992d62001/jy992d62001k_3.pdf, pp. 424–427.