A Study on Real-Time Control System with FPGA

Graduate advisor: Shuichi ICHIKAWA 013749 Hiroshi YAMAMOTO

1 Introduction

Programmable logic controllers (PLCs) are widely adopted for sequence control of industrial machinery. However, the performance limitation of PLC is recently becoming a serious problem, where higher precision and larger scale control are required. Another problem is that PLC program is rather easy to duplicate and to analyze. This often results in the leakage of valuable trade secrets and the rise of clone products in market.

Recently, it has been proposed to implement control programs in hard-wired logic by using reconfigurable devices such as FPGA (Field Programmable Gate Array), which can be arbitrarily re-programmed in actual application fields. An FPGA chip contains maximally ten million logic gates, and thus can contain a very large control system in a small chip. This may lead to reducing the number of components, which may lower the cost. Therefore, the application of FPGA technology is not limited to a large-scale applications that require high performance, but it is also applicable to small systems. It should be also noted that FPGA technology is better than PLC in protecting intellectual properties and trade secrets, because it is more difficult to analyze an FPGA hardware than to analyze a PLC software.

This study describes the implementation of control logic library, which is required to integrate a whole control system in a FPGA chip. A sample implementation of FPGA control system is also presented.

2 Methodology

PLC programs are often described in ladder diagram, which originated from relay circuit and thus can be implemented in logic circuit [1]. However, a modern PLC is a kind of computer, and it is not so easy to implement an actual PLC program in hardware.

Though there are some studies to translate PLC program into hardware [2], it is not all-powerful. Even if PLC program could be implemented in an FPGA, it does not work without other peripheral devices. Hence, it is practically essential to prepare a library of peripheral devices and popular control primitives for FPGA implementation. The author implemented several functions for control library, but only one device is introduced here.

Pulse generator (PG) is a peripheral device to drive a stepping motor. PG include many functions, but they are not always necessary simultaneously. Therefore, PG was divided and implemented in two separate modules. The first module (STPG) generates pulses of a designated frequency with linear acceleration and de-acceleration (Fig. 1). STPG adds or subtracts a designated value from the current frequency at a designated period until the current frequency reaches at terminal frequency.

The second function (TRPG) receives input pulses and send out pulses multiplied by a designated constant n/m, where $n \leq m$ (Fig. 2). TRPG increments *accumulator1* by n when an edge was detected. If accumulator1 gets

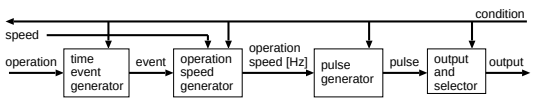

Figure 1: Block diagram of STPG

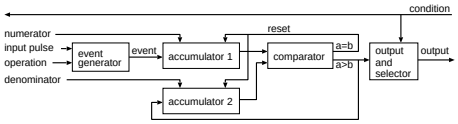

Figure 2: Block diagram of TRPG

greater than *accumulator2*, then TRPG turns over the output and increments accumulator2 by m. If accumulator1 and accumulator2 are equal, reset both to zero. This algorithm is similar to DDA (Digital Differential Analyzer) in computer graphics.

3 Implementation

Ichikawa laboratory developed an experimental winder in a joint research project with Yashima Netsugaku Co. Ltd. In this study, the author implemented the simplified control logic of this experimental winder on an FPGA board, where STPG and TRPG are integrated in FPGA. Figure 3 illustrates the block diagram of this experimental FPGA control system.

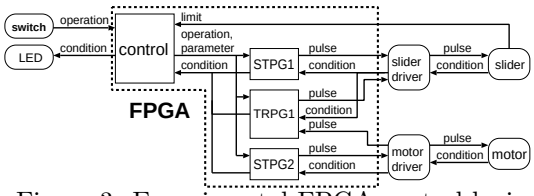

Figure 3: Experimental FPGA control logic

In Fig. 3, STPG2 drives a motor, which includes an encoder and returns pulses that are synchronized to its rotation. TRPG1 receives these pulses, and outputs pulses multiplied according to its own settings. A linear slider is driven by the pulses from TRPG1. STPG1 is used for the initial positioning of linear slider.

The abovementioned logic was described in VHDL and implemented on an FPGA by using Altera QuartusII 4.0 software for an EP20K200E device. The derived circuit costs 430 LE (logic elements) in logic scale, which is only 5% of an EP20K200 device. The maximal operational frequency was 50.38 MHz, which was higher than the onboard system clock (33.3 MHz). This circuit was downloaded on an FPGA board, and a motor and a slider successfully operated without any trouble.

4 Conclusion

An example of FPGA control logic was actually implemented and successfully operated. This would be the first step to the application of FPGA to control circuits. Since the functions of control logic library are not yet satisfactory, they have to be extended still more. Though this study examined a small control system, larger systems should be examined and evaluated on FPGA chips.

References

[1] Miyazawa, I. et al.: Implementation of Ladder Diagram for Programmable Controller Using FPGA, *Proc. 7th IEEE Int'l Conf. Emerging Technologies and Factory Automaton (EFTA'99)*, Vol. 2, pp. 1381–1385 (1999).

[2] Ikeda, R. and Ichikawa, S.: Preliminary study on H/W translation of PLC program, Proc. IEE Gen. Conf. 2005, (to appear).