Building a Configurator to Optimize Soft Processors for FPGA

Tetsuo Miyauchi (1310708)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 7, 2015

Keywords: FPGA, Verilog HDL, CPU, MIPS, configurator.

Currently, using FPGA (Field Programmable Gate Array) for embedded applications is becoming popular because of its easiness of reconfiguration while FPGA is relatively lower density and slower compared with application specific integrated circuits. As FPGA is reconfigurable, that is its circuit can be modified after a product is developed, it is prevailing for embedded application due to higher productivity.

Recently, a software processor core could be implemented in FPGA due to improvement of circuit integration. In the view of resource consumption, a processor core with a smaller size is better. So, in this thesis, in order to achieve effective use of resources in FPGA, building a configurator to create an optimized soft processor for an application is described. Several application programs are applied to this configurator and the effectiveness of this configurator from the view of the resource utilization and implement timing is illustrated.

In chapter 1, the background and the method of this study are shown.

In chapter 2, several related works are overviewed. First, several soft processor cores, which FPGA vendors provide, are shown. Then ASIP(Application-domain Specific Instruction-set Processor) is introduced. ASIP is a technology for creating application specific CPU and its environment.

In chapter 3, CPU design for this study is explained. In order to verify effectiveness of this configurator, a MIPS architecture soft processor core

is designed using Verilog HDL. Firstly, implemented CPU specification is illustrated in this chapter. Secondly, forwarding and stall detection units and classification of CPU resources are described. In order to build an optimized circuit, it is necessary to clarify the resources used for each CPU instruction. Additionally, cases in which the forwarding detection unit and the stall detection unit are used in this CPU are mentioned.

In chapter 4, building the configurator for optimizing processor cores is described. Firstly, this configurator disassembles object codes for an application program and extracts instructions used in the program. Secondly, this configurator selects actually used resources such as multiplexers, stall detection units, forwarding detection units, comparison circuits, the multiplier, and the divider, according to the classification of CPU resources, as mentioned in the previous chapter, by the result of analyzing instructions which an application program uses. And also, GUI(Graphical User Interface) for this configurator is introduced. GUI is developed with Python.

In chapter 5, evaluation and consideration are discussed. In order to confirm effectiveness of this configurator, several application programs written in C language are applied to the configurator. In this thesis, results of applying the configurator to application programs of matrix product, quick sort and SHA1 are shown. These programs are executed on the CPU explained in chapter 3, and it is confirmed that the correct result is obtained for each program. Resource utilization and implemented timing are evaluated for the circuit which the configurator created for each application program. The evaluation results are shown in the table below.

Table 1: CPU result

	Matrix	Qsort	SHA1	Full
Register	653	589	593	738
LUT	1202	1795	1814	9269
Slice	399	555	569	3238
IO	167	167	167	167
DSP48A1	8	-	-	8
Minimum Period(ns)	13.040	11.870	10.964	13.920
Maximum Frequency(MHz)	76.687	84.246	91.358	71.839

From these results, compared with a full implemented CPU, the circuit

which this configurator created is found to be effective from the view of resource utilization and implemented timing. And also, this configurator is effective for reduction of human resources in developing optimized soft processors.

In chapter 6, CPU design, configurator development, and the result of evaluation are summarized and this thesis is concluded.