DYNAMIC RECONFIGURATION OF FPGA DEVICES

Tamás RAIKOVICH
Advisor: Béla FEHÉR

I. Introduction

Nowadays, systems built with FPGA devices are becoming more and more widely used. Their great advantage is their flexibility that arises from their programmable nature as compared to systems using application specific integrated circuits (ASICs). In addition to conventional applications, the modern programmable logic devices appear also in the scope of high-performance computational structures. The reconfigurable, application specific computational architecture and the availability of large quantity of resources enable to achieve much better efficiency compared to the conventional solutions. When general purpose processors and reconfigurable computational structures are used together, the most advantageous properties of both components can be utilized. Algorithms (searching, sorting, signal and image processing, etc.) that are implemented such way have achieved 100 times or 1000 times better performance.

Using reconfiguration, designers can dramatically increase the functionality of a single FPGA, allowing a system to be implemented with fewer and smaller devices than is otherwise required.

This paper introduces a simple reconfigurable system and focuses on the Xilinx Virtex FPGA families and Xilinx development software.

II. Configuration of the FPGA devices

The FPGA devices provide a number of serial and parallel configuration interfaces that can be used to download the configuration data to the device. During the normal configuration process, the operation of the FPGA is suspended, the existing configuration is cleared, the new configuration is loaded into the configuration memory and then the device is restarted.

There are FPGA devices that support partial dynamic reconfiguration. During the partial dynamic configuration process, the device remains fully functional while a given part of the FPGA is reconfigured.

The Xilinx Virtex-II, Virtex-II Pro, Virtex-4 and Virtex-5 FPGAs have a special configuration interface called ICAP (Internal Configuration Access Port). Using this internal hardware module, the downloaded application itself can reconfigure a part of the FPGA device [1].

III. A simple reconfigurable system

This design is implemented on the Avnet Xilinx Virtex-4 FX Evaluation Kit. The board contains a Xilinx XC4VFX12-FF668 (Virtex-4 FX) FPGA and it has UART and 10/100 Mbps Ethernet communication interfaces. Because of its higher speed, the Ethernet interface is used to communicate with the PC.

Figure 1 shows the block diagram of the design. Although the Virtex-4 FX devices contain embedded PowerPC processor, still the MicroBlaze soft processor was chosen. This way, the design can be easily ported to the high-performance Virtex-5 FPGA devices, which don't have embedded PowerPC processors. The LMB ICAP peripheral connects the ICAP to the data side LMB (Local Memory Bus) of the MicroBlaze processor. The LMB was chosen here because the write and read cycles on the LMB are one clock cycle length. The OPB EthernetLite is necessary for the Ethernet communication. The OPB timer has two timer registers. One timer is required by the TCP/IP protocol, the other timer is used to measure the reconfiguration duration. The system clock frequency is 100 MHz, which is the maximum allowed clock frequency of the ICAP.

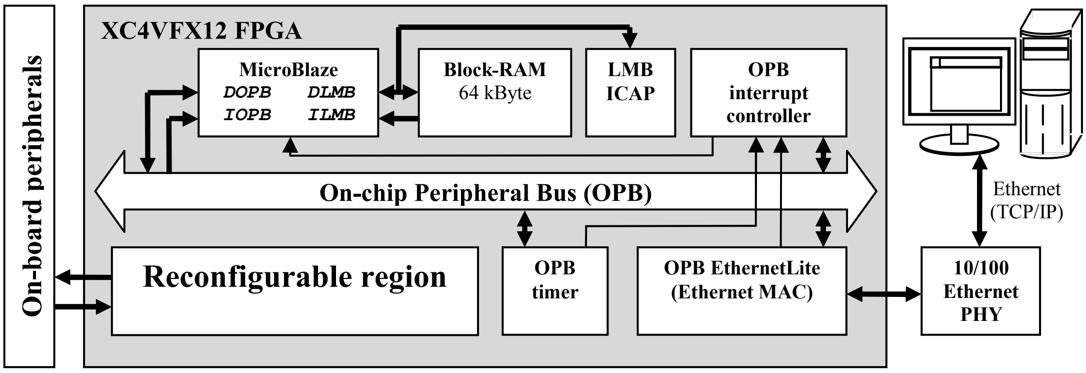

Figure 1: The block diagram of the reconfigurable system

Like normal systems, partially reconfigurable systems are also written in VHDL or Verilog hardware description language. However, these kinds of designs require much more development effort and special considerations [2]. A partially reconfigurable system consists of three main parts: the static module, the partially reconfigurable modules and the top module. Each module must be implemented in a separate project and the final merge step will generate the full and partial configuration files. The static module contains those parts of the design that won't be changed during the reconfiguration process. The partially reconfigurable modules contain those parts of the design that can be replaced with each other. All partially reconfigurable modules for a given partially reconfigurable region must be pin compatible with each other. Connections between the partially reconfigurable modules and the base design must go through a bus macro, which locks the routing between the partially reconfigurable modules and the base design. The size of the partially reconfigurable regions is determined by the minimal reconfigurable area of the given FPGA device. Lower-level modules cannot contain any clock or reset related FPGA primitives (for example BUFG, DCM or STARTUP_VIRTEX4) or I/O buffers. The top module must only contain the black-box instantiation of the lower-level modules and it also instantiates the bus macros. Xilinx ISE 9.1.02i_PR2, Xilinx EDK 9.1 and Xilinx PlanAhead 9.2 softwares were used to implement this hardware.

The software for the MicroBlaze processor uses the uIP 1.0 TCP/IP stack. As for the PC side, a simple command-line application is available to partially reconfigure the FPGA.

A future application of this design is to create a high-performance reconfigurable search engine, which performs searches in large databases. Such algorithm is used for example in medicine research where large number of molecule descriptors have to be compared with each other.

IV. Results

In this design, the partial configuration bitstream size is 23508 bytes. Using the LMB ICAP peripheral, the reconfiguration duration was 0.47 ms at 100 MHz system clock frequency. The full configuration bitstream size of the XC4VFX12 device is 595696 bytes. If the reconfiguration of the whole FPGA was assumed, the reconfiguration process would take 12 milliseconds. Consequently, an algorithm can be inefficient, if it requires many reconfigurations.

References

[1] Virtex-4 Configuration Guide, Xilinx, Inc.
URL: http://www.xilinx.com/support/documentation/user_guides/ug071.pdf

[2] Early Access Partial Reconfiguration User Guide, Xilinx, Inc.
URL: http://www.xilinx.com/support/prealounge/protected/docs/ug208.pdf