DISTRIBUTED RECONFIGURABLE DIGITAL SYSTEMS VIRTUAL LABORATORY

Dragos Popescu, Mihnea Rosu-Hamzescu,
Adrian-Cristian Petrescu

Abstract

In this paper we will explain how to remotely use an FPGA development board as if it was right on your desk.

In order to be able to remotely use the board we need two things: to be able to configure the device and to send/receive data to/from it.

Our goal is to design a building block for a virtual laboratory using some commercially available boards. We will describe the means to access a number of Xilinx test boards so students can test their own FPGA projects.

Keywords: *FPGA, virtual laboratory, Ethernet, distributed*

2000 *Mathematics Subject Classification*: 94C99, 94-02.

1. Introduction

Field Programmable Gates Arrays enabled the advent of a new computing paradigm, based on direct hardware computational algorithms implementation. The price of this technology is rapidly decreasing, but in order to make it easier to take advantage of the new wave we need access to FPGA based boards.

A RAM based FPGA can be configured from a local proprietary FLASH PROM, by a JTAG adapter hooked to a local PC interface or any another component, which copes with one of the FPGA configuration protocols.

There are solutions that can configure and communicate with an FPGA based design, but all need to incorporate some communication protocol and

logic in the design. We would like to give students full control of an FPGA and as much I/O ports as possible.

A virtual laboratory can be easily set up by connecting multiple modules in a network (local or distributed).

2. Proposal

Our building block consists of 2 D2SB boards, 1 Spartan-3 board, 1 network enabled board and 1 interconnection kit. One module can be used by 2 independent users at once or as a more complex test board.

The user has 40 programmable I/Os to communicate to one D2SB board. The 2 D2SB boards are linked together by 96 programmable I/Os so they can easily be used as a pair.

Using the large number (40) of I/O ports and a programmable clock generated from the management board (the Spartan-3 board), the user can implement a wide variety of projects and can use the entire Spartan 2E FPGA (XC2S200E) from the D2SB board.

The Spartan-3 board is used for management (clock and I/Os). The network enabled board and the interconnection kit are designed specially for this project.

We can program the test boards via a JTAG chain and communicate to the management board so we can send/receive data over the network interface.

2.1 Remotely programming the boards

Modern Xilinx FPGAs offer eight different configuration interfaces:

1. JTAG
2. Parallel-Slave, synchronous parallel with clock to FPGA, legacy
3. Parallel-Master, synchronous parallel with clock from FPGA, legacy
4. Serial-Slave, synchronous serial with clock to FPGA, legacy
5. Serial-Master, synchronous serial with clock from FPGA, legacy
6. BPI: Byte Peripheral Interface, FPGA controls readout from parallel PROM
7. SPI: Serial Peripheral Interface, FPGA controls readout from SPI-PROM
8. Select Map; fast interface for configuration

By using a JTAG interface, Xilinx devices are easily programmed and tested without expensive hardware. Multiple devices can be daisy-chained, permitting a single four-wire Test Access Port (TAP) to control any number of Xilinx devices or other JTAG-compatible devices. The network enabled board implements the JTAG state machine and allows users to remotely configure the FPGAs using the locally generated .bit file like in the normal flow.

2.2 Communication and clock management

The network enabled board is needed to transmit/receive data to and from the management board.

The management board repacks the data and controls the clock synchronization so that the test board receives the clock and data properly. The outputs are recorded and transmitted back to the user via the network interface. As the user I/Os can be configured we use a masks and a control word with 40 bits for every sample. The results are also packed 40 bits per sample word.

2.3 Network interface board

The network interface is based on a small Ethernet MAC/PHY chip, the ENC28J60, and a 16bit microcontroller with enough program memory to accommodate a TCP/IP stack plus the user routines. Optionally, a 16Mbit FLASH chip may be added to locally store multiple configuration files. Since typical endurance is around 10.000 writes, it is recommended to use it for special purposes only.

The Ethernet interface chip and the FLASH memory chip both use a 4-wire SPI interface to the microcontroller. The Spartan-3 board is interfaced to the microcontroller using an 8 bit parallel interface that enables it to quickly send stimulus data and retrieve results.

Considering the presented methods to configure an FPGA, we have chosen JTAG for a few obvious advantages. While the protocol implementation may be a bit more difficult, the interface is compatible with most of the devices. It is also possible to daisy-chain many devices and program them with a single 4-wire bus.

A special peripheral isn't necessary for a JTAG interface, and the protocol is very suitable for a software bit-bang implementation. Data-in and state signals are sampled on the positive edge of the clock and data-out is presented on the negative edge of the clock.

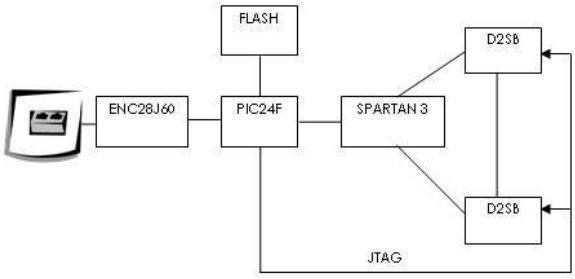

Figure 1: Simplified Module Schematic

There are only a few constraints regarding clock and data skew when the chain grows too long. Typically a JTAG chain of 20 devices is considered very long but skew can be easily prevented by inserting wait states between presenting a data/state signal on the bus and the next clock edge.

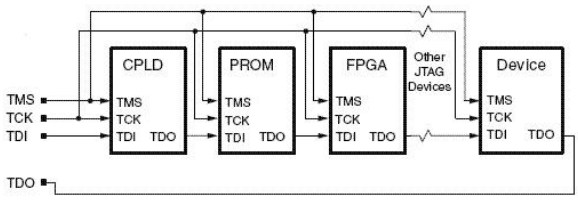

Figure 2: JTAG Interface

3. User Interface

The TCP/IP stack loaded into the microcontroller includes HTTP server functionality. This is very useful for accepting input and returning results in a platform independent manner. D2SB configuration data and stimulus files would be uploaded directly in a web interface. Since the HTTP server supports dynamically generated pages, the results may be also presented to the user in the web interface. Typically the newest result data for each board is stored in the Spartan-3 FPGA memory and is accessed by the microcontroller when needed.

A FTP server/client package may be also used for uploading files or running batch stimulus files.

4.Conclusions

An Ethernet interface offers a lot of flexibility to the virtual laboratory solution. Using commercial "all purpose" boards connected together with the Ethernet interface board also makes the solution cost effective and simple. The web based user interface makes it easy to use and attractive.

Replacing the widely used JTAG programming cable by a microcontroller, an RJ45 MagJack, an SPI-FlashPROM in a new hardware design that allows online configuration and data exchange introduces almost no extra cost neither extra design-work. The main work is to be done in the development and clever design of the server-software running on the microcontroller and hardware to seamlessly integrate it into existing tool chains. This work has to be done once, because this design is transparent to the targeted FPGA and the application under test.

References

[1] Xilinx Inc., Xilinx In-System Programming Using an Embedded Microcontroller

[2] Xilinx Inc., Spartan-3 board manual

[3] Digilent Inc., D2SB board manual

[4] Microchip Inc, PIC24FJ128GA010 Family Data Sheet

[5] Microchip Inc., ENC28J60 Stand-Alone Ethernet Controller with SPI Interface

[6] Microchip Inc., TCP/IP Stack for PIC18, PIC24, dsPIC, and PIC32 MCUs

Dragos Popescu, Mihnea Rosu-Hamzescu, Adrian-Cristian Petrescu
Department of Computer Science
University POLITEHNICA Of Bucharest
Spl. Independentei, nr. 313, 060042, Bucharest, Romania
email:{*dragos, mihnea, padrian*}*@csit-sun.pub.ro*