An FPGA-Based Experiment Platform for Hardware Software Co-Design

Yajuvendra Nagaonkar and Mark L. Manwaring

Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah 84602, USA
{yaju,manwaring}@ee.byu.edu

Microprocessor architecture and digital system design has been a focus of research and the curriculum in the Electrical and Computer Engineering Department (ECEN) at Brigham Young University. There has been continual progress in development of new and efficient architectures and custom digital systems, but these ideas have not had easy access to a design and experiment environment. There has been a need for teaching and providing practical experience to the students on the most modern concepts of computer architecture, within the time constraints of a semester. The system we are currently developing uses Field Programmable Gate Array (FPGA) devices to create a platform for designing and experimenting with microprocessor architectures and area of hardware software co-design.

In recent years FPGAs have lent themselves to the computer engineering curriculum at universities worldwide. The reprogrammable nature of FPGA make them a ideal for educational purposes as it allows students to iterate in their design tasks [1]. The use of FPGA technology often requires tools and software environments produced by the specific FPGA vendor. The costs of these can often affect the research or teaching budget at the university. Further, there is a significant learning curve in using commercial CAD tools for FPGA boards [2]. We targeted our system to minimize the use of such tools and to develop a system with a low-cost configuration environment with minimal on-board resources.

FPGA hardware systems are already available in the low end category such as the Xess board and the several boards from Digilent, and other high end boards for specific application such as the XtremeDSP Development Kit board from Nalltech. Many of these boards have restricted resources or components or include redundant components not applicable to design. These boards do not provide an adequately flexible solution for a researcher or a student implementing their design.

Our system is built upon an existing communication and power supply (CommPS) module that is currently used for our research and class room work. The CommPS has available on it different communication protocol ports like RS-232 and USB. The board is also equipped with a voltage translators so any commercial voltage supply (9v-15v) can be used. Different microprocessor and FPGA boards can be connected to this CommPS board via a 40-pin header to exploit the CommPS's communication channels for easy programmability and communication with multiple boards.

We approached the issues of FPGA configuration to minimize the cost of the system. An alternative method of configuring the FPGA by a low cost microcontroller was suggested. To implement this method, we designed two different boards, one with the micro-controller system and another one with the FPGA system. Also many of the applications of FPGA-based systems require communication with external devices such as memories,displays, microprocessors or other FPGAs. These two boards communicate with each other and the CommPS board via the 40-pin header.

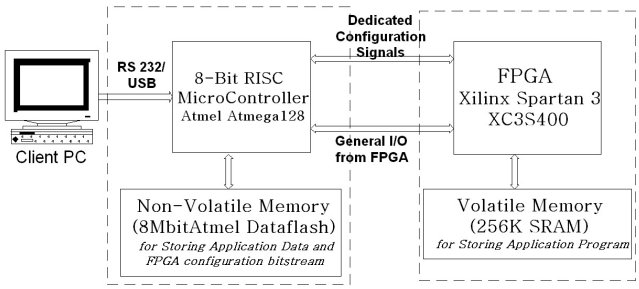

Fig. 1. The user can store the FPGA configuration data in the Flash memory and load it to the FPGA on startup. The FPGA uses on-board SRAM for storing Application Programs. After configuration the FPGA can function as a stand-alone system with communication links to the Atmel microncontroller and other peripheral devices.

The micro-controller board consists of a 8-bit RISC Microncontroller (Atmel Atmega128) and 8-Megabit Flash memory (Atmel DataFlash). The FPGA bitstream is downloaded from PC via the RS-232/USB port and can be stored in the Flash memory. The micro-controller can use the configuration data from the flash memory to program the FPGA via a slave-serial configuration method. The micro-controller board also contains voltage regulators which provide the three different power supply sources for the FPGA (Xilinx Spartan 3). Two LEDs are also provided on this board to assist with debugging the micro-controller interface.

The FPGA board consists of a Xilinx Spartan 3 FPGA and a 4Mbit SRAM. In the current version, JTAG ports are provided to debug any problems arising from the micro-controller to FPGA interface. The microcontroller and the FPGA board are compact (2.5x2.5 inches) while lending maximum functionality. A set of two 40 pin headers on FPGA board provide

communication with micro-controller board for configuration data and communication with additional FPGA boards or other peripheral device boards.

The FPGA and SRAM combination allow for emulation of processor soft-cores with different architecture features. The presence of a non-volatile memory in form of ROM, allows the system to function as a stand alone machine, as the FPGA can be configured on startup from the bitstream stored in the ROM. The system forms an unique environment, in which any custom digital machine can be constructed, utilizing the FPGA as its strength. The soft-cores of microprocessors or IP cores of machines with their own unique compiler support and software applications can be implemented on the system. The connectability of the board allows for interfacing with other peripheral devices when solving the problem of hardware software co-design.

From a classroom perspective, we project its use in computer architecture courses where development of custom machines can be realized in different experiment stages. The student is able to experiment with architecture features like bit addressing and datapath widths. The board can be easily incorporated into various research efforts. The machine vision group of our department can easily connect their custom hardware to our FPGA system to utilize soft-cores on the FPGA. The system will assist the Biomedical research groups, by creating a platform for their different signal processing and sensor design ideas. As the hardware platform for custom soft-cores, it will stimulate the area of developing IP soft cores.

Fig. 2. The Microcontroller and FPGA boards are easily stacked on the CommPS board. The CommPS board provides both RS-232 and USB ports for communication with the Client PC.

The significant advantages of our system include its compact size, low cost and abundant I/O. The system contains a minimal implementation of FPGA, necessary clocking device and SRAM on the FPGA board, and simple microcontroller and flash memory interface for FPGA configuration and three voltage regulators on the microcontroller board.This simple design allows students to easily connect any number of daughter cards using the FPGA as the switch matrix as well as a processing element. If researchers develop a specific custom board for their application, it can be connected to the FPGA board systems as well via the 40-pin header. The whole system is self contained and does not require any proprietary programming cables or any external hardware.

An example of a hardware software co-design is the problem of selecting the best soft-CPU to support application of an RF module. The RF module has a weak communication link and requires error detection and correction. The co-design problem can be approached by either moving the error detection and correction to hardware or software. If a function is implemented in hardware, the appropriate hardware module can be interfaced with the FPGA module. Another area of experimentation is with use of FPGAs from different vendors to conduct different performance studies with them.

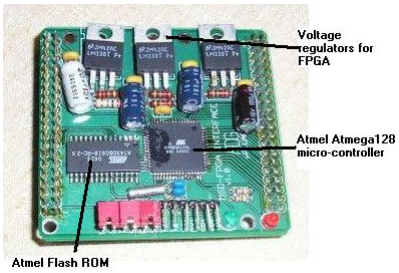

Fig. 3. Microcontroller board with a 8-bit Atmel Atmega128 and Atmel 8Mbit Dataflash. The board also has three voltage regulators to provides the three different voltages as required by the FPGA. The Microntroller is used for configuring the FPGA from the configuration data stored in the Flash memory.

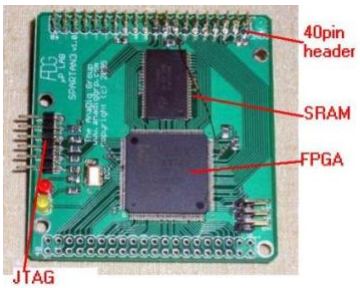

Fig. 4. FPGA board with 400K gate Xilinx XC3S400 Spartan 3 FPGA and 256KX16 (4Mbit) Cypress Semiconductor SRAM for Application Program Storage. JTAG programming option is provided for debugging the configuration process.

REFERENCES

[1] M. Holland, J. Harris, and S. Hauck, "Harnessing fpga's for computer architecture education," in *2003 International Conference on Microelectronics Systems Education (MSE'03)*, p. 12, IEEE Computer Society, 2003.

[2] T. S. Hall and J. O.Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE transactions on Education*, vol. 47, pp. 502–507, November 2004.