

ECE994 – PSoC (Programmable System on a Chip)

Global Network Academic Test (GNAT)

“A GLIMPSE INTO THE FUTURE: DESIGNING
INFRASTRUCTURE WITH AMBIENT INTELLIGENCE”



By Dale Ferriere
December 5, 2006

For ECE994 / Dr. Andrzej Rucinski and Class Project Team

1.0 Introduction

During the University of New Hampshire's 2006 fall semester, a handful of students, most with computing and electronics engineering backgrounds, and a few with mechanical engineering and marine safety and security backgrounds, took an experimental course entitled "Programmable System on a Chip" (PSoC). ECE994 is created with the idea of becoming a critical component of the Systems Engineering undergraduate and post graduate curriculum including the futuristic aspect of global ambient intelligence networks (GAIN). The course reference is, by Clive "Max" Maxfield, "The Design Warrior's Guide to FPGAs". ECE994 objectives tasked students with the following: (a) In a laboratory setting prove the application of a XILINX XUP (XILINX University Program) field programmable gate array (FPGA); and, (b) Extrapolate from the proven prototype and identify future scenarios on how FPGAs or PSoCs may be applied creating ambient intelligent infrastructure enhancing safety and security design. A secondary and more abstract course objective is to show how Systems Engineers' internal design process (i.e., writing code, developing bit streams) needs to be simplified in order to be better understood by policy makers and for other engineering disciplines.

1.1 What is an FGPA?

A field programmable gate array is a semiconductor device containing programmable logic components and programmable interconnects. The programmable logic components can be programmed to duplicate the functionality of basic logic gates such as AND, OR XOR, NOT or more complex combinational functions such as decoders or simple math functions. In most FPGAs, these programmable logic components (or logic blocks, in FPGA parlance) also include memory elements, which may be simple flip-flops or more complete blocks of memories.¹

1.2 How does the FPGA work?

A hierarchy of programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer, somewhat like a one-chip programmable bread-board. These logic blocks and interconnects can be programmed after the manufacturing process by the customer / designer (hence the term "field programmable") so that the FPGA can perform whatever logical function is needed.²

Although current generation FPGAs are generally slower than their application-specific integrated circuit (ASIC) counterparts, can't handle as complex a design, and in earlier designs drew more power, they have several advantages over the ASIC such as a shorter time to market, ability to re-program in the field, and lower non-recurring engineering costs. In contrast to the ASIC, the FPGA, which contains a power PC, is advantageous because the FPGA can be field configured for specific functions or purposes, such as processing sensor inputs. These advantages offer a glimpse at future scenarios when FPGA applications are included with the design, maintenance and operation of critical infrastructure.³

1.3 Current Uses

In his article, “FPGA: Coming to a PC Near You”, Phil Bishop, president and CEO, CeloxicaPhil Bishop, discussed how AMD and DRC Corporations demonstrated an accelerated computing system using Celoxica’s C-programming environment to couple an Opteron processor with a dynamically reconfigurable FPGA-based coprocessor module. “This is earth-shaking news if you understand FPGAs. And even if you don’t, you soon will.” FPGA co-processing is the most exciting shift in computing thinking since Marcian “Ted” Hoff dreamed up the processor. The exploitation of the inherently parallel nature of the huge FPGA input / output bandwidth by implementing C-based code directly as Register Transfer Level (RTL) gates.⁴

The following are some other examples. HiEnergy Technologies halved the false positive rate with no increase in the interrogation time on its systems to detect explosives, biological agents, or illicit drugs using C-based FPGAs. HiEnergy uses C-based design and synthesis tools to optimize the algorithms and synthesize them directly to FPGA hardware.⁵

Calsonic Kanei Corp. doubled the performance of collision avoidance systems based on number plate recognition by exploiting the potential of low-power high-performance, flexible FPGAs to solve the complexity involved and the unique constraints of in-car electronics. Combining parallel reconfigurable architectures with design entry and implementation, automotive designers can move into new silicon paradigms and quickly implement systems that out perform traditional solutions in terms of power, performance and time to market.⁶

AppServer Solutions realized an FPGA coprocessor for its real-time hyper-spectral imaging movie processor in less than one month. HIS is used in defense and civilian remote sensing tasks, such as chemical and biological weapons detection, underground bomb-damage assessment, foliage penetration, mineral exploration, and environmental monitoring.⁷

2.0 PROVING THE CONCEPT or GET SOMETHING WORKING

At the onset of the ECE 994 course, one of the course’s primary objectives was to gain hands-on experience with an FPGA. An FPGA manufacturer, XILINX provided the University of New Hampshire with their XILINX XUP for the purpose of providing students with hands-on experience. Some students initially thought that the FPGA could be quickly adapted and applied to a specific project associated with their doctoral program. The class quickly determined, because no one had experience in building code for an FPGA, for the time allotted this was not feasible. It was agreed to keep the approach simple (Quote) Get something working (Unquote). After open discussion, a project team was established and challenged to configure a XILINX XUP FPGA that could, on a table-top environment, detect for the presence of a coffee cup “intruder”, and, for extra credit, determine if the contents of the coffee cup was either hot or cold.

2.1 Homeland Security and the Project Team

Dr. Rucinski challenged the students to think of how an FPGA, because of its singular focus, ability to be configured in the field, and capability to network sensor inputs, could be used for homeland security type scenarios. A project team, consisting of six persons and, based on their backgrounds, strengths and desires, were assigned to the following positions: Project Leader; Project Documentation Leader; Designers (two-persons); Web-Master; and, Scenario Writer. The suggested prototype problem statement included being able to configure the XILINX XUP FPGA to receive and recognize analog sensor input signals (e.g., thermometer, infrared camera, and/or accelerometers) and show outputs indicating when there was a change in baseline condition. All agreed to what was thought to be at that time a reasonable design objective, not knowing the significant challenges associated with developing and implementing code (bit stream, kernel and operating system) for the XILINX XUP FPGA.

2.2 FPGA Workshops

To accommodate student needs, two workshops were established for working with the XILINX XUP FPGA. First, a private workshop was operated from a student's residential home located in the greater Boston area. Those working from this home-based workshop previously purchased XILINX XUP FPGAs and components. Second, the remaining students would use the University of New Hampshire micro-electronics lab. To integrate and enhance learning between the two workshops, and create a virtual workshop where students, although physically located at separate locations, could work together, the University of New Hampshire's LearnLinc (www.learnlinc.unh.edu) medium was successfully used.

2.3 Course Meetings

By the third course meeting all students had performed the built-in self test (BIST). The BIST for the XILINX XUP FPGA is broken down into two categories: (1.0) Processor Tests; and, (2.0) Hardware Tests. The following processor tests were tried: EMAC Web Server, which enables the FPGA to link with the Internet and provides opportunities to communicate data (inputs / outputs) through and from the world wide web; AC97 Audio Test, which enables the FPGA to process audio inputs and outputs; MGT SATA; System ACE; and DDR SDRAM. The EMAC Server and AC97 Audio tests were successful. The other processor tests were initially unsuccessful because of missing components e.g., SDRAM, A/D Inputs / Output board. The following hardware tests were tried and were successful: Power supply, reset test; Clock, push button, DIP switch, LED, and Audio Amp Test; and, SVGA Gray Scale Test.

Additionally, students received several applicable lectures. These lectures included initial presentations on "Microelectronics Design Domain, differentiating between technology, architecture and design methodologies. During this lecture students were made aware of the Dajski Diagram, which depicts a target separated by three axes (Behavioral, Structural, Physical / Geometry) with Architectural spheres from which the Systems Engineering design process flows outward. Also, Dr. Kochanski provided several lectures on Networking Sensors and "Interfacing the Dirty World of Analog to the Ideal Purity of Digital". Personal tutoring was provided for those with limited

electronics engineering backgrounds on basic microelectronics. Tomasz Jankowski presented a lecture downloaded from the XILINX University Program entitled, “EDK Base System Builder (BSB) support for the XUPV2P Board”. This lecture made students aware of digital signal processing, Aurora Internet Protocol and included discussions on filters for incoming signal processing, developing bit streams, connecting XILINX blocks, multiplier accumulator, XILINX Core Generator and synthesizing (placement and routing). Representatives from MATHWORKS presented a lecture and demonstration of their design methodology for the FPGA on how to use MATHLAB - SIMULINK to develop signal processing and communications code and language. Dr. Rucinski provided a lecture on the history and development of VHDL (very high speed integrated circuit hardware design language).

2.4 Lesson Learned

Following the BIST students realized, because no one on the project team had previous experience with writing code, there were looming challenges associated with FPGA design methodology. Three design methodologies were studied for potential use in writing code for the XILINX XUP FPGA or XUPVSP Board: XILINX blocks; Matlab – Simulink; and, Mentor Graphics. Because the FPGA being used was manufactured by XILINX the thought was that their design methodology would be easier to use. However, creating bit streams and kernels proved to be, for the experience level in the classroom, overwhelming. At one of the final workshops, at which Mr. Sam Raynolds was a guest lecture, commenting on the challenges associated with developing code for the FPGA, “Uncertain and hard to deal with.” Students realized, because of not knowing how to write and apply code to a XILINX XUP FPGA, that the course objective of configuring an FPGA to be able to interface with two separate analog input nodes was not feasible. A new prototype aim was established to get two XILINX XUP FPGAs to talk to each other, perhaps between the two workshops.

3.0 Future Applications

Being able to configure a system on a chip in the field or field programmable gate array has tremendous potential aiding public and private sector policy makers, managers, and operators at preventing potential terrorist incidents and minimizing or preventing other incidents. For instance, the ability to dynamically configure networked sensor inputs, making infrastructure and vehicles intelligent, could be applied to the following scenarios.

- (a) Create smart tunnels and smart mines – strategically placed network of vibration sensors could forewarn operators and laborers in real-time about potential structural failures (see Big Dig catastrophe) and make early detection of fire, smoke and explosive atmospheres.
- (b) Create intelligent cargo containers – in-container atmosphere and physical sensors could be dynamically-field-configured by using bill of lading and cargo manifest information to establish an in-container’s baseline physical-chemical property expectations (expected in-container atmospheric, gravitational forces and cargo off-gas sensor expectations) thereby effectively detecting for in-container

anomalies, eliminating false positives and being able to communicate anomalies directly to first responders.

- (c) Create intelligent commercial ships – network of sensors placed at strategic points of a cargo ships structure thereby forewarning operators in real-time when stresses and strains exceed acceptable limits; and, collision avoidance sensors integrated with radar and mooring systems thereby aiding operators in their efforts to avoid collisions.
- (d) Create intelligent highways and railways – network of sensors placed along interstate highways and railways detecting for chemical, biological, radiological, nuclear, and explosives, contrast inputs with data-bases of expected hazardous material transports and trains thereby forewarning officials about potential anomalies.
- (e) Create intelligent shipping channels – network of sensors placed at choke points detecting for chemical, biological, radiological, nuclear, and explosives, contrast inputs with data-bases of expected hazardous material shipments, thereby forewarning officials about potential anomalies. Moreover, speed detectors could be placed along the shipping channel making officials aware of small boats and other seaborne traffic exceeding local harbor standards; this application may provide advance warning about a potential USS Cole style attack.
- (f) Create smart schools – network of sensors placed around the perimeter of a school and integrated with geo-fence technology thereby creating safe and secure zones, detecting for the presence of weapons and targeted for would be predators, required to wear corresponding RFID tags.

4.0 Conclusion

Experience gained from working with the XILINX XUP FPGA was invaluable. Students received a glimpse at the future of what Programmable Systems on a Chip could have on creating ambient intelligent infrastructure. However, with this developing technology, in order to realize the potential of an FPGA, manufacturers should work to simplify design methodology so that the challenge of developing operating code is straightforward and requires less trial and error.

References:

1. Design Warrior's Guide To FPGAs, by Clive "Max" Maxfield, pp. 1-49, copyright Mentor Graphics Corporation and Xilinx 2004, <http://en.wikipedia.org/wiki/Wikipedia>, ECE 994 Lecture Series
2. ibid
3. ibid
4. <http://www.chipdesignmag.com/display.php>, "FPGA: Coming to a PC Near You", by Phil Bishop, November 3, 2006
5. ibid
6. ibid
7. <http://news-info.wustl.edu/tips/page/normal>, "Sensor networks protect containers, navigate robots, by Tony Fitzpatrick, November 9, 2006