A Versatile FPGA Demonstration Platform for Academic Use

Uli Kretzschmar, Julen Gomez-Cornejo, Naiara Moreira, Unai Bidarte, Armando Astarloa, *Member, IEEE*
Department of Electronics and Telecommunications
University of the Basque Country
Bilbao, Spain
{uli.kretzschmar, julen.gcb, naiara.moreira, unai.bidarte, armando.astarloa}@ehu.es

***Abstract*—As technology evolves and electronic systems become increasingly more complex it is important to attract students for technical studies. Our group has developed a multifunctional prototype in order to overcome the initial reservation towards high-end digital logic. This prototype gives young students possessing over only a small technological background an attractive introduction into this field. Besides this school event type of utilization of this prototype, it is also designed to be the basis of a variety of project based learning in the academic context.**

***Index Terms*—FPGA, academic platform, school-event demonstrator, System-on-Chip**

I. Introduction

The feasibility of utilizing FPGA in undergraduate studies has been shown in different previous works [1], [2]. In this context it is important to partition and restrict the workload of the students in order to avoid frustrations based on the increasing complexity of electronic systems. Evaluation platforms provided by FPGA vendors like Altera and Xilinx (e.g. [3], [4]) focus on the engineers, which already possess knowledge and skills in the field of hardware and software development. Students on the other hand might experience some difficulties in working through the corresponding documentation. Other sources of difficulties can be the partitioning of soft- and hardware together with the general high level approach towards a specific educational problem.

In recent years FPGA have grown to be a potential platform for a wide variety of applications. Whereas first FPGAs were typically used to implement glue logic between the main components of a system, today's FPGAs enable the realization of very complex systems. They offer high system operation frequencies, a high amount of available resources and a significantly lower Non Recurring Engineering (NRE) cost than ASICs. An additional trend are FPGAs, which implement reconfigurable logic together with a hard wired processor core. This way it is possible to efficiently implement SoCs, which combine the benefits of a fast and compact general purpose processor and application specific modules for efficient parallel data pre-processing. Specialized features such as partial reconfiguration allow FPGAs to be used in a unique way changing parts of the implemented logic, while other parts continue to be functional.

Our group has worked on a versatile prototype to introduce students into this aspiring technology. This prototype is not only focused on providing singular technological skills, but on combining multiple fields of studies forming an attractive final result. A highly modular architecture allows straight forward partitioning of individual implementation tasks and the collaboration of different student groups even from different studies.

The remainder of this work is structured as follows: Section II presents the basis for our prototype, the evaluation board ML507. In the following Section III the objectives we want to achieve by the prototype are presented. Section IV discusses the hardware, programmable hardware and software modules necessary to achieve these objectives. In the last Section V our first use case of the prototype is presented before the paper is concluded in Section VI.

II. Basis of the Prototype

The main architecture of the developed prototype utilizes the FPGA development board ML507 provided by Xilinx [4]. A block diagram of the components available on this board is illustrated in Figure 1.

This platform includes a huge variety of output devices, such as a VGA/DVI output, a 16x32 character LED, multiple debugging LEDs, a piezo speaker and an AC97 device for sophisticated sound processing. Several push-buttons, switches and an incremental encoder can be used as direct inputs. Furthermore, the ML507 board also comprises of a variety of external interfacing possibilities, such as USB, Ethernet and RS232. Finally many different clock sources and memories conclude the basis for a broad range of applications.

The center of the previously described peripherals on the ML507 board is build by the FPGA device XC5VFX70T. This FPGA from the Virtex-5 FX platform, which is specialized for embedded processing, contains one PowerPC processor block. Using this hard wired processor it is possible to construct complex Systems-On-Chip (SoC) without losing FPGA resources for the CPU part. Other research fields with an elevated interest for academic projects are the partial programmability and the availability of DSP48E slices, which are specialized logic resources on the FPGA for digital signal processing tasks.

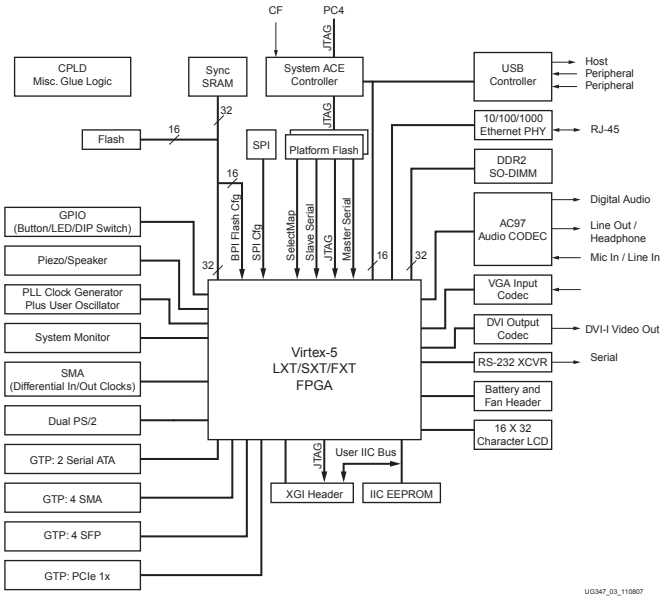

Fig. 1. Block diagram of the ML507 evaluation board

III. Objectives of the Prototype

The main objective of the prototype is to educate students using state of the art devices and design tools. Generally it is feasible to implement student projects on a relatively low level of complexity, which still can be handled by students in its entity. We nevertheless found that such strongly simplified systems induce less motivation and poorer learning results as compared to more elaborated ones.

More precisely, we aimed at implementing a base architecture on top of the ML507, which will be called *prototype* from this point on, which addresses the following usage areas:

1) *Development of programmable hardware modules:* Typical SoCs implement a main processor together with a variety of peripherals. Specialized application requirements need to be addressed by the development of a specific peripheral. Examples for these modules are co-processors, filter algorithms or communication modules.
2) *Embedded software development:* The focus of the embedded software will be the development on the PowerPC using the industry standard tool Eclipse, which is included in the Xilinx toolkit. Nevertheless, other forms of embedded software developments can also be addressed, as for example the usage of the PicoBlaze processor [5] for implementing complex state machines or an educational processor architecture such as YASP [6].
3) *Development of external hardware:* In addition to the development of modules on the FPGA, we also want to address the development of external hardware extensions, which can be connected to the prototype. Examples of these external boards are external inputs such as keypads and joysticks, external LED arrays as output display units and even H-bridges for motor control.
4) *Demonstrators for scientific contributions:* One important application field of our prototype is the visual demonstration of scientific results in the context of research projects. Most of the successful publications do produce a great amount of theoretical evaluations and measurement data, which are difficult to present in an appealing fashion. On this purpose the prototype will serve as a demonstrator, which can also be used in "open day" events to attract new students to study a degree in the electronic field.

All use cases offer the possibility of developing different laboratories of different complexities. The development of peripheral SoC modules can be as basic as implementing a state machine for a rotary encoder. Or it can be a very complex task such as the implementation of a hardware co-processor for complex filter algorithms. There is also a lot of flexibility in the part of the embedded software development. As an alternative of using the hard wired PowerPC, the soft processor MicroBlaze can be used, as it is completely supported by the Xilinx toolkit. Assembler programming exercises can be addressed by implementing functionality with tiny processors such as the PicoBlaze.

These four different objectives result in a potential for students with different backgrounds. Telecommunication and electrical engineering students can work in each of the different aspects of the prototype. Computer science students can be integrated in the embedded software development and power electronic students might be able to use the prototypes external output features for control tasks. One very interesting addition we envision for the prototype is to support remote laboratories [7] where the students can test their implementations from their home computers.

IV. Structure of the Prototypes Interfaces and Implementation

At this moment different features of the prototype are already implemented and available and others are only targeted but not yet supported. Figure 2 illustrates the available features in dark grey boxes with continuous lines, whereas future features are illustrated using light grey boxes with dashed lines.

A. Features of the current Prototype

For the first version of our prototype, all functionality for basic user interaction has been implemented.

The prototypes main output terminal is a VGA/DVI monitor output. On the ML507 board a Chrontel DVI transmitter device is implemented for generating the DVI and VGA output signals. The DVI transmitter is connected to a control module on the FPGA, which continuously sends the contents of the video memory. This module and the required DDR2 memory interface are implemented using the Xilinx EDK and due to its complexity we do not plan any student work on this part of the prototype. As an input method for the prototype the five push buttons named *Cursor* in Figure 2 can be used. All

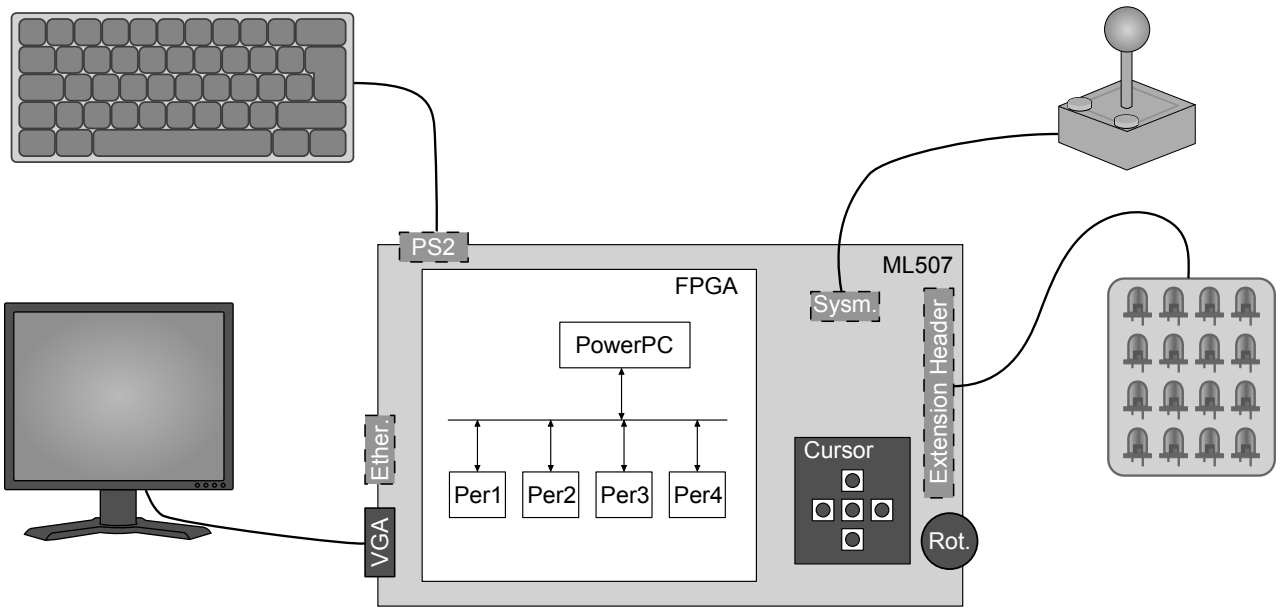

Fig. 2. Prototype structural block diagram and external extentions.

buttons create interrupts for each push. These cursor buttons are a good example to introduce this method of handling user input/output operations. The cursor buttons and the previously mentioned VGA output represent the minimal implementation set for the demonstrator on which all student works will be based.

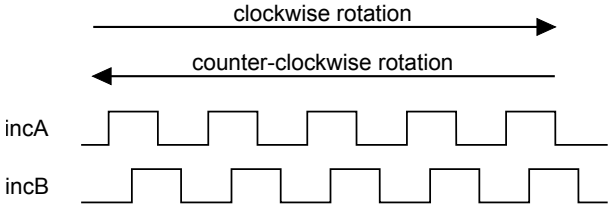

Fig. 3. Typical waveform for a rotary encoder.

The hardware component with the label *Rot.* in Figure 2 is a rotary encoder. A typical waveform is illustrated in Figure 3. Both output signals *incA* and *incB* are connected directly to FPGA I/O pins. The prototype already disposes of a peripheral module for accessing the rotary encoder. Nevertheless, this hardware is ideal to execute basic student works on the implementation of simple state machines, or for demonstrating the differences of interrupt- or polling-based input handling.

From the software side of the basic prototype features, the DVI/VGA output represents the main area of interest. The data for the video output resides in the DDR2 memory of the ML507 platform in form of a two-dimensional array, which holds an integer holding the 24 bit color information for each pixel. The current software comprises of different methods for the following tasks:

- *Drawing lines:* An automated form of drawing lines between two points using the Bresenham-algorithm [8]. Using this function more complex figures based on lines, such as polygons, can be implemented in a straight forward fashion.
- *Drawing circles and ellipses:* The Bresenham algorithm also allows an efficient implementation of painting a circle or ellipse into the video memory.
- *Writing text:* Based on a user font, which is stored in the FPGAs internal memory a function allows to write text and other symbols into the video memory.

All these elementary functions allow outputting data in a structured way and represent a minimal set of functions, which is required for attractive video outputs.

B. Extensions to the current Prototype

Clearly, the prototype presented until this point represents only a basic feature set. This was required to achieve the requirements for our first utilization of the prototype in a open day event, which will be presented in Section V. In addition to these basic functions more advanced features are planned to be implemented. Some of these implementation tasks are well suited to be executed by students in a hands-on training.

On example of this is the graphics library. It will be extended by functions, such as the following:

- *Filling objects:* The algorithm flood fill allows to fill any object starting from a singular point until a border line is reached.
- *Scalable text:* This function extends the already implemented text output function by including a parameter containing the desired text output size.
- *Rotation matrix:* This slightly more advanced method allows to rotate certain pixel coordinates to achieve a more flexible graphic output.

Especially the functionality of rotation matrices is interesting because it could be accelerated by the implementation

of an external peripheral module implementing this complex function.

In addition to the current input methods (cursor and rotary encoder) it is interesting to add two further methods, a keyboard and an analog joystick.

- *PS2 keyboard:* Adding a PS2 keyboard is supported by the EDK toolkit by Xilinx. This makes the addition easy and a potential subject for a student laboratory.
- *Analog joystick:* The addition of an analog joystick is also an interesting student lab. It requires the implementation of an external circuitry with operational amplifiers to adapt the voltage levels to the FPGA I/O and the utilization of the FPGAs AD converters, which are part of the system monitor.

An additional possible output resource are multiple LEDs, which can be connected to the extension header as indicated in Figure 2 for the example of a 4x4 LED matrix. Such a module requires the development of a SoC peripheral, which can be programmed to output different patterns in the PowerPC software.

V. Application case of the Prototype

After having presented the prototype's general structure in the previous section, this section will briefly describe the first use case in which our group was able to apply the prototype: a demonstrator of a scientific work.

A. Research Background of the Demonstrator

One focus of our research group is the implementation of fault tolerant applications. Faults can occur when high energy particles enter the silicon substrate of the FPGA die [9]. On their way through the silicon substrate these particles generate an ionizing trail, which can cause glitches or memory upsets. Since FPGA implement a great amount of configuration memory to hold the information about the implemented design, memory upsets are very critical to the correct function of the implementation.

In this context one interesting branch is the combination of Dynamic Partial Reconfiguration (DPR) and Triple Modular Redundancy (TMR). An example of this combination is illustrated in Figure 4.

The critical implementation module is tripled and all inputs are fed to all three instances. A subsequent majority voter receives the processed data of the three redundant modules. In a process called *voting* all inputs are compared and the singular value is forwarded to the voters output, where at least two redundant modules coincide. The logic for implementing a one-bit voter is illustrated in Figure 5. In case of a correct value of "0" for all inputs *A*, *B* and *C*, any false "1" on a single input will be filtered by the AND gates. In the opposite case of a correct value of "1" for the inputs, a false "0" on one of them will result in one AND gate, which still lets the correct value of "1" pass to the OR gate. A TMR setup with voting allows the complete design to continue operating, even

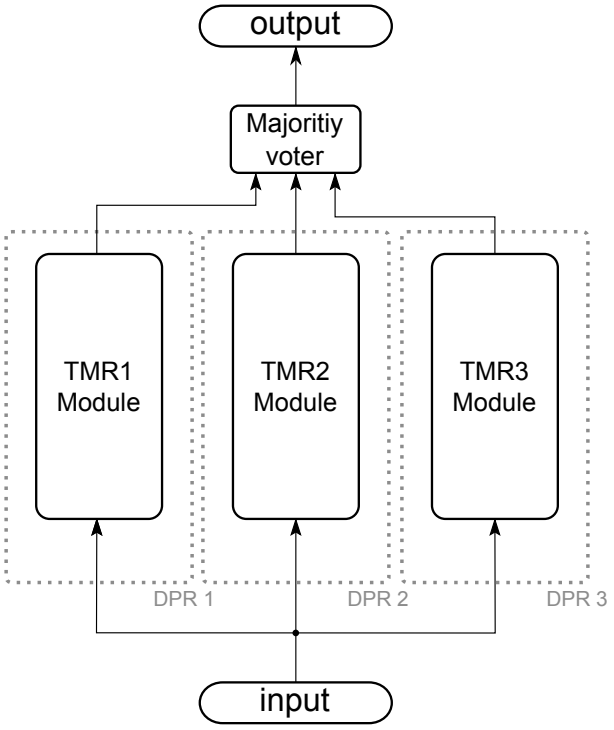

Fig. 4. Block diagram of an autonomous fault tolerant system.

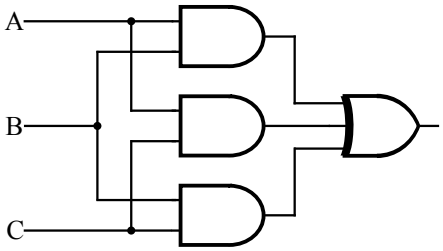

Fig. 5. Implementation of a one-bit voter.

if one of the triplicated modules is damaged by a radiation induced SEE [10].

DPR, which is illustrated as indicated by the dotted rectangles in Figure 4, augments the previously described fault masking capability of TMR by fault reparation. If the TMR-voter detects one module to be faulty then the corresponding triplicated module needs to be repaired. In this step the SEE affected module is reconfigured partially and by this any configuration upset is repaired, while the other two modules keep being operational.

This method works well for many different applications. But if the module, which is to be protected against faults, contains a complex internal state the pure combination of TMR and DPR is no longer sufficient. One typical example for such a module is a processor. In case one processor suffers a SEE and is afterwards repaired by DPR, then the repaired processor will start from its reset state and thus have a different internal state than the other three.

To address this issue we proposed four different synchro-

nization methods with different complexities and synchronization completenesses in an anterior scientific work.

B. Aim of the Demonstrator

Based on different synchronization methods we have developed in a previous work we implemented a demonstrator, which itself is based on the prototype, focusing on the following aspects:

1) Emulation of SEE by injecting errors in the individual TMR modules.
2) Demonstration of the fault masking functionality of the TMR voter.
3) Comparison of two different synchronization methods.

By implementing these three features the resulting demonstrator is able to show the insights of a completely functional design based on TMR and DPR. After the injection of a SEE in a first step it is possible to observe the voted system output, which will still be correct. After repairing the error in the faulty instance and a subsequent synchronization step, the system will go back to its fault free state.

C. Details on the Demonstrator

In order to provide an interesting demonstrator, which could also be used in open day events, we decided to implement the worlds first video game PONG. The game is illustrated in

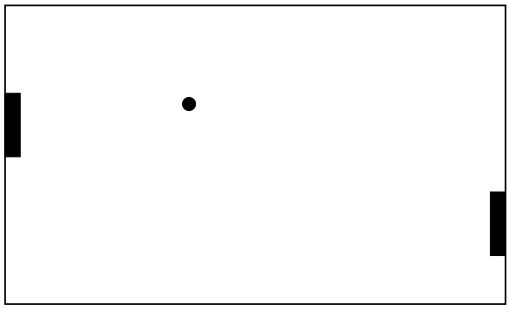

Fig. 6. The game of PONG.

Figure 6 and consists of two paddles and a ball to play an elementary form of tennis. The actual game is realized on the tiny processor core PicoBlaze and the whole design is tripled and prepared to be partially reconfigurable as presented in the previous section.

The TMR protected setup was implemented in a single peripheral module of our academic prototype. All the three PicoBlaze processors receive the data from the rotary encoder, which controls the movement of the right paddle in the game of PONG, the players paddle. The movements of the ball and the computer paddle are calculated by the PicoBlaze processors. All processors write the positional data of the ball and the paddles to the interface registers of the peripheral. In this way software on the PowerPC of the prototype can read this data of the three PicoBlazes and execute the voting on the processor.

A picture of the demonstrator is shown in Figure 7. The output on the monitor is structured into three areas:

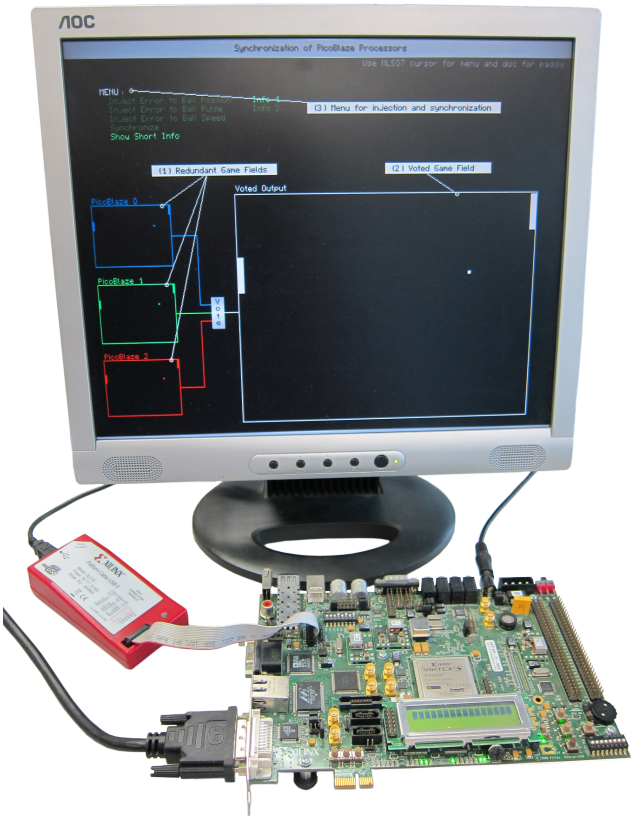

Fig. 7. Usage of the prototype in a scientific demonstrator.

The first area on the left hand side of the monitor contains illustrations of the three individual PicoBlaze processors, which all run the game of PONG. Each individual game field for each redundant PicoBlaze processor is illustrated using a different color. The second area on the right of the monitor contains the voted game field. And finally on the top of the screen a menu is shown, which can be controlled using the cursor buttons of the prototype. Using this menu it is possible to inject different kinds of errors into any of the three PicoBlaze processors. The different synchronization methods can also be triggered here.

Using this demonstrator it is possible to inject errors, observe errors in single redundant parts of the game but not the voted game field and finally repair the error. We did successfully exhibit this demonstrator at the *XIII Semana de la Ciencia, la Tecnología y la innovación* in order to present research activities to young students in order to attract them to our field of studies.

VI. Conclusion

In this paper we have presented our efforts to implement an academic prototype based on the ML507 evaluation board by Xilinx. Our aim was to generate a general purpose architecture, which can be used for many different academic aspects. These aspects can be the implementation of student laboratories in different study fields, but also the demonstration of publication ideas in an appealing fashion. By providing a

basic programmable hardware and software structure on top of an existing evaluation board, it is now possible to enable students to work on attractive applications. In addition to this we presented our first application of this prototype for the demonstration of one scientific publication of our group. This demonstration was successfully presented at the *XIII Semana de la Ciencia, la Tecnología y la innovación*.

The unimplemented parts of the prototype represent our main focus for future works together with the generation of documentation for the successful execution of student works and laboratories utilizing the prototype.

ACKNOWLEDGEMENTS

This work has been carried out inside de Research and Education Unit UFI11/16 of the UPV/EHU and supported by the Department of Education, Universities and Research of the Basque Government within the fund for research groups of the Basque university system IT394-10 and by the Ministerio de Ciencia e Innovación of Spain within the project TEC2011-28250-C02-01.

REFERENCES

[1] K. Nakano and Y. Ito, "Processor, Assembler, and Compiler Design Education Using an FPGA," in *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, de. 2008, pp. 723–728.

[2] H. Ochi, "ASAver.1: an FPGA-based education board for computer architecture/system design," in *Design Automation Conference, 1997. Proceedings of the ASP-DAC '97 Asia and South Pacific*, jan. 1997, pp. 157–165.

[3] Altera Corp., "Stratix V GX FPGA Development Board Reference Manual," Altera Documentation, MNL-01063 http://www.altera.com, jul. 2012.

[4] Xilinx Corp, "ML505/506/507 Evaluation Platform User Guide," Xilinx Documentation, http://www.xilinx.com, oct. 2009.

[5] K. Chapman, "PicoBlaze 8-Bit Microcontroller for Virtex-E and Spartan-II/IIE Devices," Xilinx Documentation, XAPP213 http://www.xilinx.com, feb. 2003.

[6] L. Ribas-Xirgo, "Yet Another Simple Processor (YASP) for Introductory Courses on Computer Architecture," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 10, pp. 3317–3323, oct. 2010.

[7] I. Santana, M. Ferre, E. Izaguirre, R. Aracil, and L. Hernandez, "Remote Laboratories for Education and Research Purposes in Automatic Control Systems," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 547–556, feb. 2013.

[8] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, apr. 1965.

[9] Xilinx Corp., "Considerations Surrounding Single Event Effects in FPGAs, and Processors," Xilinx Documentation, WP402, http://www.xilinx.com, mar. 2012.

[10] H. Quinn, P. Graham, M. Wirthlin, B. Pratt, K. Morgan, M. Caffrey, and J. Krone, "A Test Methodology for Determining Space Readiness of Xilinx SRAM-Based FPGA Devices and Designs," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 10, pp. 3380 – 3395, oct. 2009.