

VALID: Custom ASIC Verification and FPGA Education Platform

Patrick Murphy, J. Patrick Frantz, Erik Welsh, Ricky Hardy, Tinoosh Mohsenin and Joseph Cavallaro
Rice University

Abstract

This paper describes VALID, a platform for testing student designed ASICs and for teaching the basics of FPGA design. VALID is designed to maximize ease of use from a student's perspective while maintaining enough flexibility for its use as an FPGA development and instruction platform. This system was designed entirely by students, has been successfully manufactured and is currently being used in a number of courses at Rice.

1. Introduction

VALID is a hardware and software platform with two primary purposes. The first is the functional verification of custom ASICs designed by students in Rice's undergraduate VLSI course. Second is its more general use as a platform for teaching the basics of FPGA design and implementation.

The following sections include a description of VALID's hardware and software, followed by a discussion of its applications in educational environments. We close with some concluding remarks and acknowledgements.

2. Hardware Overview

In order to accommodate its two primary uses, VALID includes a large number of peripherals and external interfaces. The peripherals include a LCD character display, audio CODEC and on board SDRAM. The external interfaces include a socket for the custom ASIC and a significant number of general digital inputs and outputs. VALID also includes a pair of Mictor connectors connected to the ASIC socket for use with an external logic analyzer. Although one goal of this project was to eliminate the need for such dedicated test equipment, we felt it was important to maintain compatibility with existing solutions. This interface also offers an additional means of troubleshooting problems in both ASIC testing and FPGA design projects.

At the heart of the system, however, are two major components: an Axis Etrax 100LX microprocessor [1] and a Xilinx Virtex FPGA. These components are housed on separate custom boards, allowing a great deal of future expandability and maximal reuse of previous designs. VALID was designed specifically to allow the Etrax processor to configure the FPGA, the benefits of which are discussed below.

A. Axis Etrax 100LX Microprocessor

VALID's microprocessor is a 100 MIPS RISC processor by Axis Communications. The Etrax is designed specifically to run Linux, the open-source Unix-like operation system. This is an extremely useful feature as it greatly simplifies the software development process. The development environment is based on the standard GNU compiler, linker and debugger, allowing seamless porting of existing Linux code to the Etrax platform. Linux support is also useful in that the source code for the entire kernel is freely available. This was important in our design given our need to configure an FPGA via the Etrax's external memory interface.

We chose a new version of the Etrax 100LX, marketed by Axis as the Etrax multi-chip module (MCM). The Etrax MCM includes the standard Etrax 100LX processor core plus integrated SDRAM, flash memory and an Ethernet interface in one BGA package. This integration greatly simplifies our hardware design, reducing the complexity of the circuit board and minimizing the number of separate components.

B. Xilinx Virtex FPGA

VALID supports any Xilinx Virtex or Virtex-E series of FPGA in a BG-560 package. Our initial versions use the Virtex XCV1000 part, an FPGA with approximately one million gates. While not the largest available, this FPGA is large enough to achieve the primary goals of this system. It

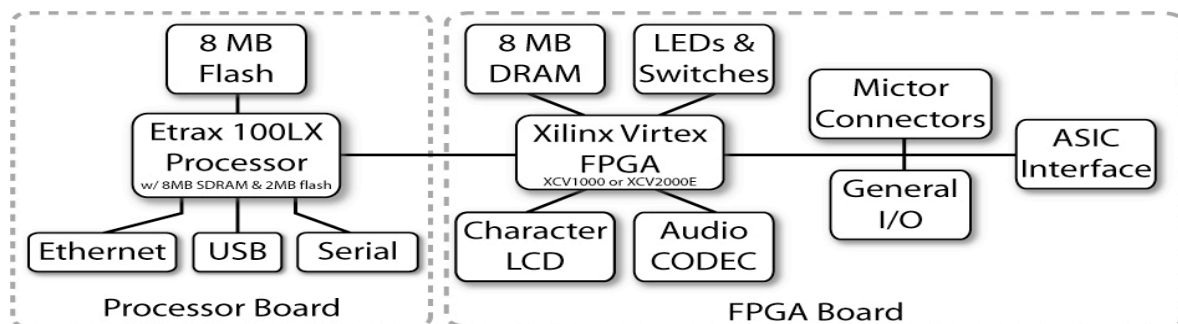


Figure 1: VALID block diagram

provides sufficient logic and internal memory for fairly complicated designs, including complex test suites for ASIC verification and intermediate to advanced FPGA design projects.

The FPGA has a dedicated 16-bit bus connected to the Etrax processor's memory interface. With the interface running at the maximum 50 MHz, this allows a theoretical throughput 8 Gbps between the devices. This significant capacity will prove useful in two ways. First, it will allow large amounts of data to be passed between an ASIC under test and the Etrax microprocessor. Second, this throughput will allow sustained data transfer with the FPGA itself. Even the smallest compatible FPGA includes enough logic for complex signal processing designs, and this high capacity, low latency link should provide sufficient throughput for exchanging data in real-time.

3. Software Overview

VALID's software implements two major interfaces. The first interface allows data exchange between the Etrax processor and the FPGA. The code implementing this interface must allow the Etrax to both configure the FPGA and exchange data with it after configuration. This software is written in C and cross-compiled to the Etrax platform. FPGA configuration operates in SelectMAP mode, the newest and fastest configuration method available in Virtex parts. After configuration, the Etrax processor treats the FPGA as a standard asynchronous memory device. We chose this design approach as it allows any code capable of accessing the Etrax memory interface to exchange data with the FPGA.

The second interface implemented by the VALID software allows the exchange of data with a client PC via the Etrax's network interface. Specifically, data can be exchanged with a Mathworks Simulink [2] model in real-time. We chose to implement an interface with Simulink in order to utilize its existing data generation and analysis features. We expect this feature will prove especially useful in FPGA projects involving signal processing tasks where the audio and general I/O interfaces to the FPGA are too slow or narrow. Additionally, we are currently investigating the development of a similar interface to National Instrument's LabView software [3].

4. VALID as a Teaching Platform

The VALID project was born from the need for a more sophisticated system for testing custom ASICs. In years past, students used very basic logic analyzers for testing their chips. As the available IC processes advanced and the complexity of designs increased, the need arose for a better testing system.

VALID's suitability as a system for teaching the basics of FPGA design became clear as the design progressed. Like many other FPGA development boards, VALID includes a large number of peripherals and I/O. It offers three significant advantages, however. First, VALID is designed for use with

FPGAs with more than one million gates. The FPGAs found in systems designed for basic and intermediate FPGA instruction are generally smaller, potentially limiting the complexity of realizable designs. Second, VALID is entirely self-contained. No external cables or any other specialized hardware is required for FPGA configuration. The Etrax processor, acting as an embedded Linux PC, needs only a bitstream to configure the FPGA. Bitstreams can be downloaded to the Etrax via FTP or NFS. Configuration can then be triggered automatically or manually via remote login to the Etrax. Where most other FPGA boards require configuration by special programming cables or pods, VALID needs only a network connection. Finally, the flash memory on VALID's processor board offers enough non-volatile storage space for many different bitstreams. Different groups of students can store their FPGA designs on the same system, or a single project can utilize multiple configurations.

A final aspect of VALID of particular interest in educational environments is its tight integration with Simulink. The utility of this integration is enhanced by Xilinx's System Generator, a Simulink blockset capable of creating FPGA designs from system models. With VALID, it is possible to create a system model in Simulink including an FPGA component using System Generator. The system can then be simulated behaviorally in Simulink. If implementation in a real FPGA is desired, System Generator can generate a bitstream for the FPGA component of the model. Using this bitstream, our software can configure VALID's FPGA and exchange data with it in real-time, entirely within Simulink. This capability offers a very intuitive environment for working with FPGAs without requiring knowledge of VHDL or Verilog. We expect this feature will prove very useful to instructors looking to introduce the basics of FPGA design to students not familiar with an HDL.

5. Conclusions

We have designed and built VALID, a hardware and software platform for ASIC testing and teaching FPGA design. This system is already being used by students to test custom ASICs designed as course projects, and its use as an FPGA education platform is anticipated in upcoming courses.

Acknowledgements

Special thanks to Xilinx, especially Anna Acevedo of the University Relations Program. Their support and donations of software and FPGA parts have made this project possible. Thanks also to Erik Swanson and Jeff Gavornik for their contributions to VALID's hardware design.

References

- [1] <http://developer.axis.com/>
- [2] <http://www.mathworks.com/products/simulink/>
- [3] <http://www.ni.com/labview/>