On the Introduction of Reconfigurable Hardware into Computer Architecture Education

Ross Brennan and Michael Manzke
Trinity College Dublin
Ireland
rbrennan@tcd.ie, michael.manzke@cs.tcd.ie

Abstract

The introduction of reconfigurable logic devices as teaching-aids, into undergraduate and graduate education, enables the students to conduct experiments they could otherwise not perform. Furthermore, this approach gives instructors the freedom to choose architectures that are dictated by the underlying hardware to a lesser extent.

Today's Field Programmable Gate Arrays (FPGA) can implement integer units as complex as the SPARC V8 at a reasonable hardware price. Educational boards that replace conventional CPUs with reconfigurable logic devices can be integrated into an existing syllabus with legacy hardware requirements without disruption, as long as the soft-CPU *core on the reconfigurable logic device provides opcode compatibility with the superseded processing unit.*

1 Introduction

For almost 20 years computer science students and computer engineering students at Trinity College Dublin have been asked to construct a complete microprocessor system as part of the coursework. The students perform this tasked by wire-wrapping integrated circuit (IC) components to a Motorola MC68008 microprocessor. The 8 bit data bus keeps the required amount of wire wrapping to an acceptable level. The CPU has no caches and therefore allows the observation of all memory accesses on the system bus. The CPU is clocked at a modest 7.5MHz. Despite its age the processor is very suitable for this task but it has become increasingly difficult to source these CPUs. We had to look for an alternative that is equally suitable for the application.

Replacing the MC68008 with a more modern CPU that exhibits similar properties is a feasible design option. We compared this option with an approach that utilises soft-CPUs that execute on reconfigurable logic devices. FPGAs that are large enough to hold a complex CPU can be bought at a competitive price.

It was obvious that an FPGA based solution would not only allow us to replicate the properties of the exiting board but also to increase the scope to an extent that it could be useful as an educational-aid for many more undergraduate and postgraduate courses.

In this paper, we argue for the migration from standard CPUs on computer architecture lab boards, to an FPGA based approach that employs *soft-CPUs*. These FPGA based boards should allow the student to download a choice of *soft-CPU* cores into the SRAM based FPGA. A number of configuration PROMs can be used to hold a range of these *soft-CPU* cores, that may be individually selected through a *PROM-select-switch* before the board is powered on. This allows the utilization of the boards for a variety of teaching objectives, from a wire-wrapped microprocessor project to CPU design projects.

We propose the following *soft-CPU* options:

- A simple instruction set processor that can be implemented by a second-year student
- An open-source, fully configurable SPARC V8 architecture conforming to the IEEE-P1754 standard
- Legacy architectures that simplify the transition to FPGA-based boards. At Trinity College Dublin (TCD) this requires a Motorola MC68008 compatible core.

All of the possible *soft-CPU* options must use a common system-bus interface. In particular, the open source SPARC core gives students access to a HDL model of a 'state-of-the-art', industry standard processor. The ability to configure the HDL sources

enables the students to activate and deactivate components in the designs in order to perform specific experiments. This will be of interest to students in advanced undergraduate and postgraduate courses.

In lower undergraduate courses, the board may be used by second year students to wire-wrap a complete microprocessor system and to design, test and synthesise a simple instruction set computer.

As a first step to evaluate these design objective we prototyped an FPGA based board that executes an open-source SPARC core (the LEON P-1754) [1] in a Xilinx XC2V1000 Virtex-II FPGA. This core was written as a highly configurable, fully synthesisable VHDL model and was originally developed by the European Space Agency (ESA) with the intention of being used in future space missions. The HDL model is configured with an 8-bit data bus and is clocked at 6MHz to enable students to wire-wrap a complete microprocessor system. The processor is configured to run with the internal caches disabled, which allows students to observe all of the bus transactions using a logic state analyser connected to the external system bus. This facilitates debugging of the system and allows the development of a rudimentary operating system.

2 Background and Related Work

Using custom hardware and reconfigurable logic devices to aid in the teaching of computer architecture is not a new concept and its success has already been demonstrated through several different projects. These projects focus mainly on teaching the students using a fixed architecture with a fixed project board configuration and thus do not take full advantage of the reconfigurable nature of the underlying hardware system.

Work has been carried out to design and implement custom hardware and simulation tools at the University of Waikato [2]. This involved developing and implementing a computer architecture solely for the purposes of teaching and then designing a project board and simulation tools to complement it.

Several efforts have been made to design processors that are not compatible with commercial Instruction Set Architectures (ISAs), such as the work carried out at Harvard [3]. Their *Ant-32* processor architecture was developed with the intention of providing a simple yet functional platform through which to introduce students to the operation of a basic microprocessor system.

A project kit based around a dedicated soft-CPU architecture has also been developed by Gray Research [4], with the intention of teaching basic computer architecture concepts to undergraduate students.

Without major updates to the processor models, these implementations do not allow the system to grow in complexity as the students understanding of system design concepts increases. By using fully functional, configurable processor models, the complexity of operation of the microprocessor system may be tailored to suit the needs of different student groups.

Currently computer architecture education in the Computer Science degree at TCD accounts for 35% of the overall teaching effort. In the first year of their studies, students are educated in Assembly Language Programming, Digital Logic Design and Electrotechnology.

These courses serve as the foundation for the Computer Architecture and Digital Electronics courses in the second year. The first semester of this Computer Architecture course requires students to construct a working microprocessor system from integrated circuit (IC) components. Students work in small groups and use wire wrapping to interconnect the required circuitry. Figure 1 shows a lab board that is currently used by students for their microprocessor project and Figure 2 provides a schematic for the same project. A basic operating system that enables communication with a host system and the execution of application code is also developed. In the subsequent semester students are introduced to VHDL and learn about register transfers, the design of a datapath, sequencing and control. As part of their coursework, students are asked to design a Multiple-Cycle Microprogrammed Instruction Set Computer. This design is implemented in VHDL.

More advanced aspects of computer architecture are covered in Computer Architecture II and Computer Engineering in third year. This includes CISC and RISC architectures, memory hierarchy, I/O subsystems, high performance processing systems and VLSI Design.

In their final year students may select two computer engineering related subjects, Computer Architecture and Integrated and Systems Design. The option is also available to students to choose a final year project with a computer architecture content.

A similar syllabus is taught to computer engineering students. Engineering students may choose to join the degree programme in computer engineering, following the first two years that are common to all engineering students. These students receive

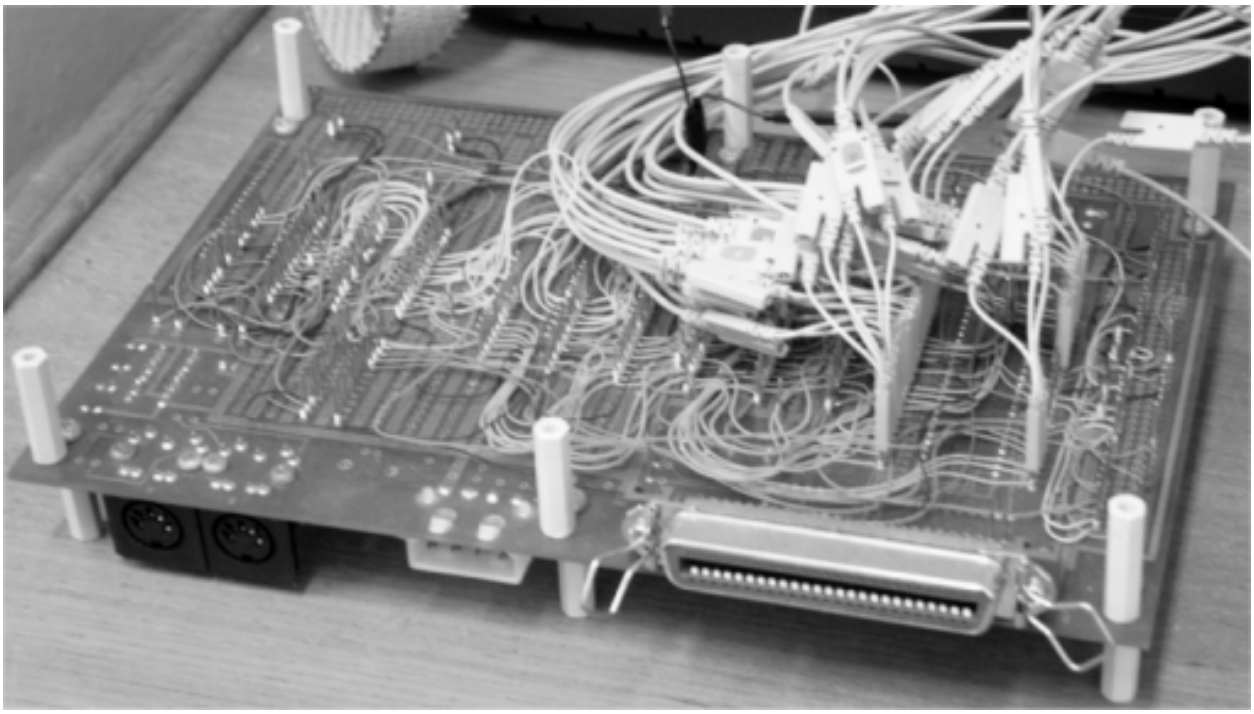

Figure 1: Photo of the current wire-wrapping lab board

the same computer architecture lectures that computer science students receive but in a compressed form over the last two years of their degree.

The proposed lab boards should be suitable for use in most of the computer architecture related subjects for both computer science and computer engineering students.

It is envisioned that the custom built Motorola 68k microprocessor system will be replaced with the new FPGA based solution. Currently the 68k microprocessor system is use to teach first year computer science students and third year computer engineering students 68k assembly language. The FPGA boards could be used in conjunction with a 68k HDL model and a daughter board that substitutes the wire-wrapped part of the board. The daughter board could be populated with more sophisticated integrated circuit components that also take advantage of a full 32bit bus width.

The daughter board could also be used with a version of the LEON core that is configured for this purpose. The core would operate with a 32bit system bus and caches would be enabled. Furthermore, there is scope for I/O capabilities such as PCI and Ethernet. This configuration is suitable for more advanced classes.

The proposed board would also allow students to construct a microprocessor system through wire wrapping by removing the daughter board and using an 8-bit version of the LEON core, the 68k compatible model or a student-built Multiple-Cycle Microprogrammed Instruction Set computer with an 8-bit system bus interface.

The current design project is based around the 16-bit Motorola MC68008 microprocessor, which was released in 1982. The main features of this processor are its CISC architecture and 8-bit external data bus. This processor is becoming too outdated to reflect modern computer architecture trends and so a new alternative had to be sought, which was more in line with current technological developments.

The microprocessor project requires each of the student groups to complete the following tasks in order to obtain a fully functional microprocessor system:

- The first task is to verify that the project board's internal clock circuitry is operational and to generate the correct 16MHz signal.
- A clock divider is then implemented using one of the GAL devices. An 8MHz signal is used as the CPU clock frequency and a 1MHz signal is used as a reference signal for the serial port baud rate generation circuitry.
- The reset circuitry is then designed and implemented in order to debounce the reset-button

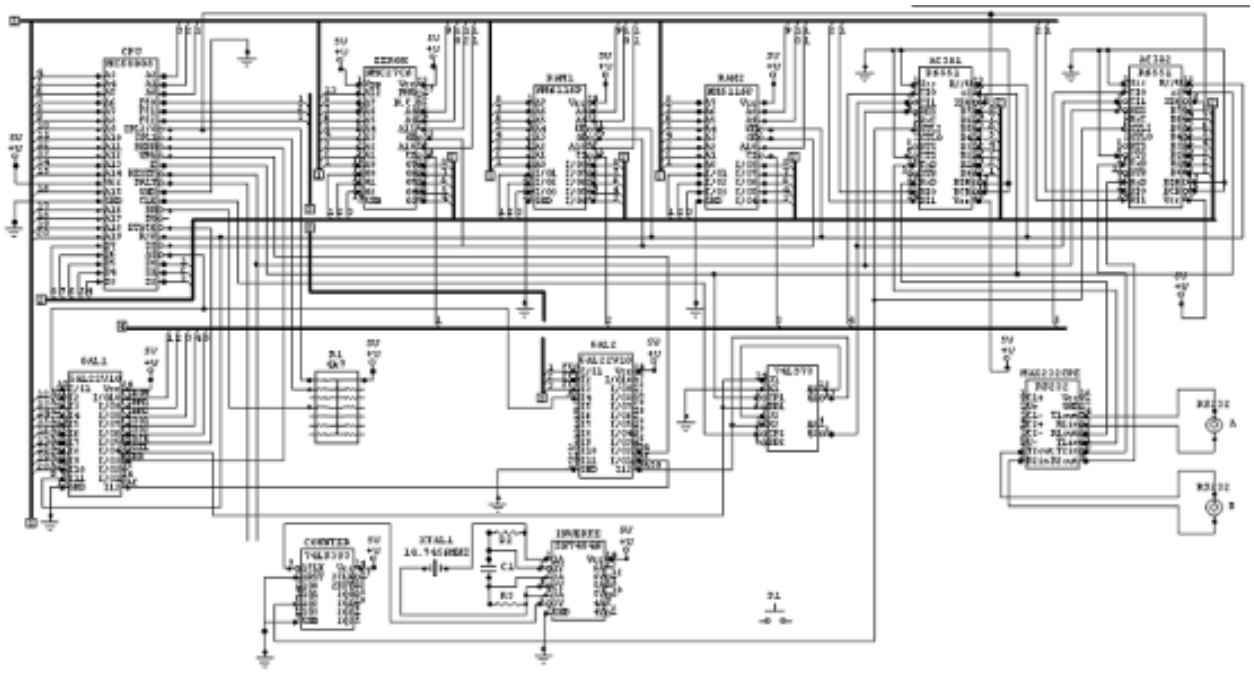

Figure 2: Schematic of the current microprocessor project

and hold the reset signal low for at least three clock cycles in order to allow the processor and peripherals to reset correctly.

- The address space is then segmented into regions where the ROM, RAMs and UARTs are mapped, using logic implemented within the GALs, in order to control the peripheral select signals according to the device memory mappings.

- The EPROM is then programmed with a test program, which allows the students to verify that the core "glue-logic" has been implemented and is functioning correctly.

- Two serial ports are then interfaced with the processor and their functionality tested using a transparent link program, which allows two separate consoles to connect to each other using the microprocessor system.

- At this stage, the project hardware has been completed and a monitor program is now written in assembly code and downloaded onto the EPROM. This is the final task in the completion of the microprocessor design project.

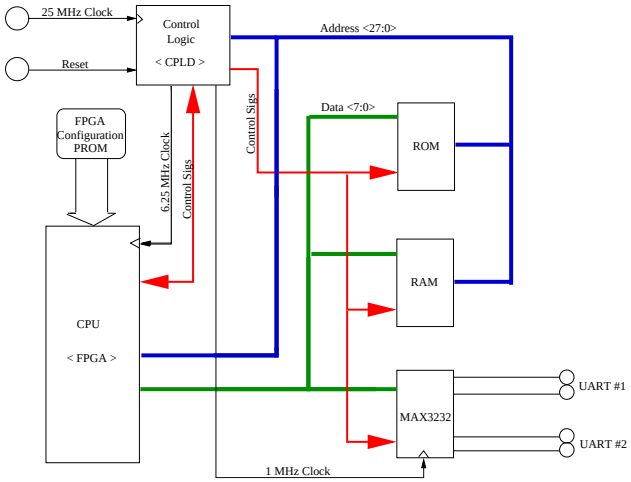

Figure 3: Block Diagram of the Prototype Design

3 The Prototype Project

The design for the new project was created using reconfigurable logic devices to implement the CPU and control logic, in conjunction with standard ROM and RAM chips, as shown in Figure 3.

For the purposes of prototyping the design, only one configuration PROM was used to store the soft-CPU. As soon as the board is powered-on, the FPGA is automatically programmed with the contents of

the PROM. This could easily be extended to provide several different PROM configurations, each containing a different soft-CPU, which could allow the user to choose the CPU option that was to be programmed into the FPGA on power-up. This could be accomplished using a switch to select between the different PROM configurations before power was applied to the project board.

The prototype hardware was designed with the intention of providing as much compatibility with the current design project hardware, in terms of layout and components, while taking advantage of the advances in technology since the advent of the original project. This influenced several design decisions, such as using LVTTL components throughout the design.

3.1 The Project Board

The hardware platform chosen to be used in prototyping the new project design was the VirtexII FPGA [5] [6] used in conjunction with the VirtexII prototyping board [7] [8]. All of the prototype project circuitry was implemented on the breakout area of the board using LVTTL components which were wire-wrapped together. One ROM chip [9], one RAM chip [10] and two serial ports were added as external devices to prove the operability of the CPU core.

All of the control logic was implemented on the same FPGA that held the CPU core, in order to prove that the design would work. Students undertaking the project, however, will implement the required control logic in a separate FPGA or CPLD to the one containing the CPU.

A standardised bus interface was developed in order to ensure that the hardware design would allow different soft-CPU options to be run, without having to change any physical wiring on the project board. Any processor that implemented this interface could be programmed onto the FPGA and run on the hardware system. It was decided to base the bus interface on that of the Motorola MC68008 processor as this was already in use in the current design project and would facilitate the migration to the new project design.

3.2 The LEON Model

The LEON-P1754 microprocessor [1] is a VHDL model of a 32-bit RISC processor conforming to the IEEE-1754 standard, which is fully compatible with the SPARC V8 reference architecture [11]. The model is fully synthesisable and can be implemented on both ASICs and FPGAs. It incorporates

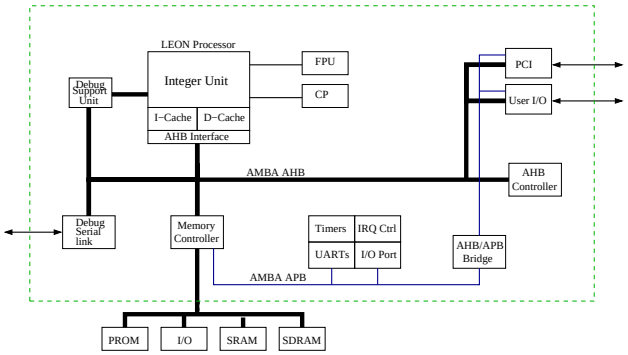

Figure 4: Block Diagram of the LEON core

an integer unit, seperate instruction and data caches and peripheral modules, which are all interconnected via a full implementation of an AMBA APB/AHB bus[12], as shown in Figure 4.

The LEON processor was originally designed by Jiri Gaisler while working for the European Space Agency and is currently under development by Gaisler Research. The first release of the LEON core was made available in October 1999 and has been continuously upgraded and enhanced since then. In order to promote the SPARC standard and enable development of system-on-chip (SOC) devices using SPARC cores, the ESA made the full source code for the processor freely available under the GNU-LGPL license.

This processor[1] was chosen as an alternative to the Motorola MC68008 due to its high level of configurability and proven operability. The processor model, as well as all of its supporting software tools, are freely available from Gaisler Research [2].

The primary method of configuring the LEON core is through the provided graphical configuration utility, which is based around the Linux kernel configuration utility. This method provides options to configure most of the model functionality. However, several major alterations were made to the model source code in order to introduce greater configurability into the model. This was important as the model had to be altered in such a way as to make it compatible with the layout of the current design project.

- The internal data and instruction caches were made removable
- The two internal UARTs were made removable

[1] LEON2-1.0.10-xst was used in the project
[2] http://www.gaisler.com

- The internal reset signal generator was made removable
- The internal memory map was altered to allow the removal of the pre-defined ROM and RAM device mappings
- The internal generation of the bus transaction signalling was altered so that the signals could be suppressed
- A new bus transaction protocol, modelled on the MC68008 protocol, was developed and implemented as a configurable option

All of these changes were implemented in such a way so that their functionality could be enabled or disabled using the graphical configuration utility. In this way, it was possible to obtain the original functionality of the model by setting the appropriate configuration options. This would allow the core to be tailored to specific levels of complexity, depending on the target group of students.

With the core configured to its most basic level, it would be suitable for use as a direct replacement for the Motorola chip in the design project. It would also be possible to include more advanced options in the core, such as the internal UARTs and caches, if a higher level of complexity was required. This means that the processor could be introduced to students at a first and second year level and could be increased in complexity in order to follow them through third and fourth year as new concepts in computer architecture, such as caching, are introduced to them.

3.3 Software Tools

Software tools for use with the LEON core are freely provided by Gaisler Research. These include a simulator, debugger and compiler, which can be used in conjunction with the LEON processor.

LECCS is the cross-compiler system for LEON. It can be used to compile C or assembly programs into a format suitable for running on the LEON core.

TSIM is a simulator that emulates the LEON core, which is useful for teaching the students about the SPARCV8 RISC assembly language and allowing them to compile and run programs on a standard pc.

DSUMON is the Debug Support Unit MONitor available from Gaisler Research, suitable for use with the LEON core. It allows access to the contents of the LEON on-chip registers when the DSU option is enabled in the processor and communicates with the board using a dedicated UART.

4 Conclusions

The prototype design successfully proved that the modified LEON core is suitable for use as a soft-CPU option on a reconfigurable hardware platform. Due to the implementation of a standardized bus interface within the LEON core, the soft-CPU was successfully able to run on the prototyped board using test programs that had been written in SPARC assembler and compiled using the LECCs compiler for LEON. The resulting hardware system is fully reconfigurable in nature and as a result gives rise to the possibility of implementing and running several different soft-CPU options on the prototype board, without having to make any modifications to the hardware.

The LEON core used as the soft-CPU could be synthesised with many different configurations, allowing a wide variety of hardware options to be tested on the board. Benchmark programs would allow the comparison of different processor configurations and would be useful for highlighting performance differences due to the various architectural features, such as whether cacheing is enabled or not, cache size, pipelining, etc. These physical comparisons would be of use to different student groups, depending on their level of knowledge about computer architectural concepts.

The "hands-on" nature of the project allows students to experiment with different processor architectures and configurations, aiding in the teaching process by adding a practical side to the theory and concepts being taught.

5 Future Work

The work described in this paper established the basic hardware design and component requirements of a reconfigurable hardware board using a synthesisable VHDL model of a microprocessor. The next task leading on from this work is to design and build a dedicated hardware board based on the prototype project design. This board could include the option to store either multiple configurations of the LEON core (with varying degrees of complexity) or several different processor architectures (implementing the standardized bus interface protocol) on the one board, using multiple user configuration PROMs.

Several different processor architectures could also be evaluated on the hardware system, for example, a synthesisable VHDL model of the MC68008 could be designed and implemented. The only constraints for implementing a processor architecture on the reconfigurable hardware board are that they implement

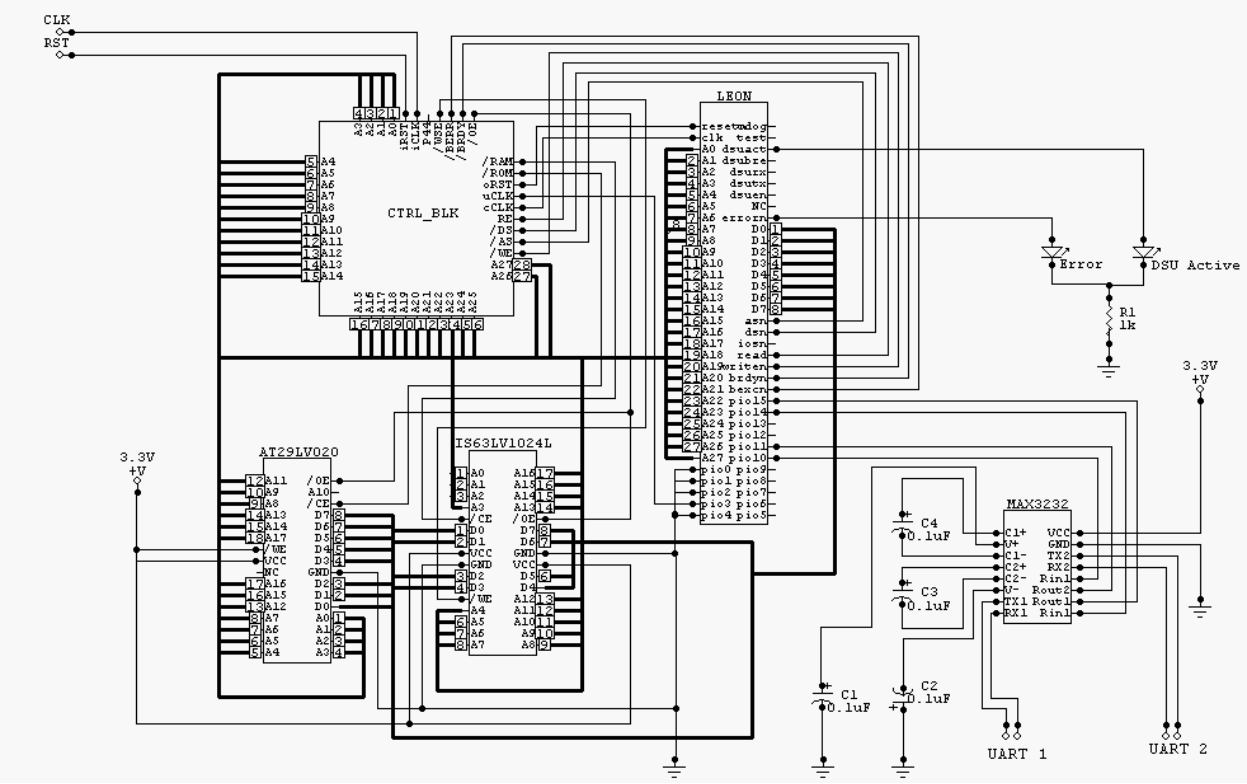

Figure 5: Schematic of the Prototype Design Project

the standardized bus interface.

References

[1] J. Gaisler, *Leon2-1.0.10 Users Guide.* Gaisler Research, December 2002. http://www.gaisler.com.

[2] Murray Pearson, Dean Armstrong, Tony McGregor, "Using Custom Hardware and Simulation to Support Computer Systems Teaching," *Workshop on Computer Architecture Education*, 2002.

[3] Daniel Ellard, Daivd Holland, Nicholas Murphy, Margo Seltzer, "On the Design of a New CPU Architecture for Pedagogical Purposes," *Workshop on Computer Architecture Education*, 2002.

[4] J. Gray, "Hands-on Computer Architecture - Teaching Processor and Integrated Systems Design with FPGAs," 2001.

[5] Xilinx, *Introduction to the VirtexII Product Family.* Xilinx Inc, December 2001. http://www.xilinx.com.

[6] Xilinx, *VirtexII Advance Product Specification.* Xilinx Inc, September 2002. http://www.xilinx.com.

[7] Xilinx, *VirtexII Prototype Platform Users Guide.* Xilinx Inc, June 2001. http://www.xilinx.com.

[8] Xillinx, *VirtexII Platform FPGA Handbook.* Xilinx Inc, December 2001. http://www.xilinx.com.

[9] Atmel, "Atmel AT29LV020 Flash Memory," tech. rep., Atmel Corporation, May 2002. http://www.atmel.com.

[10] ISSI, "ISSI IS63LV1024L Static RAM," tech. rep., Integrated Silicon Solutions Inc, July 2002. http://www.issi.com.

[11] SPARC, *SPARC V8 Manual.* SPARC International Inc, January 1992. http://www.sparc.org.

[12] ARM, *AMBA Specification V2.0.* ARM Limited, May 1999. http://www.arm.com.