AC 2012-4733: IT BLINKED! EMPOWERING STUDENTS WITH AN IMPROVED MICROPROCESSORS COURSE

Mr. Arlen Planting, Boise State University
Prof. Sin Ming Loo, Boise State University

It Blinked!
Empowering Students in an Improved Microprocessors Course

Abstract

Empowering students in understanding microprocessors involves teaching them how a processor works so that they have the skills they need when presented with a different architecture. Allowing the students to participate more fully in the discovery process enhances their ability to tackle projects with little or no help, and provides the sense of accomplishment that leads a student to exclaim "It blinked!" when he succeeds in causing an LED to blink. Soft core processors run on an FPGA development board were used to implement changes to a microprocessors course in order to achieve the desired goals. The use of soft core processors allows configuration changes not possible in traditional microprocessors.

Simplifying the processor, exposing the low level processor interactions, and adjusting the processor configuration as needed to best demonstrate the desired foundational concepts, are integral to the updated microprocessors course at Boise State University. Course enhancements, including development and continuing augmentation of a course reference, are ongoing.

Introduction

The junior microprocessors course at Boise State University (BSU) was updated in 2007 by adopting a soft core processor and adding the C programming language to teach the basics of microprocessors and peripheral interfacing techniques. The introduction of soft core processors was not the focus of the course, but their usage provides an effective, flexible platform for teaching microprocessor concepts with both the assembly and C languages. The combination of a soft core processor with a field-programmable gate array (FPGA) –based development board as the development platform, allows the course to undergo continual improvement without being limited by fixed hardware. (The Altera Nios II DE2 board was the development platform chosen for the BSU Microprocessors course.)

Though the course update was begun with a well-defined set of objectives, not all of these goals were immediately achieved. The goals have eventually been accomplished through a process of continual improvement over the past five years. This paper discusses the improvements since the initial implementation[1].

Course Goals

The goal of the junior microprocessors course at BSU is to give students an understanding of the universal foundational concepts of processors, such that they will be able to move that knowledge onto any platform in the future. To best accomplish this goal, it was vital to provide a stable course structure that would transcend changing technology yet remain up-to-date. A stable structure protects the investment of time involved in developing instructional materials. This allows the instructor to focus on improving the learning experience rather than constantly updating to follow the latest technologies, most of which are attached devices or software development tools not related to the foundational concepts.

This approach represents a departure from the typical microprocessors course, which is dependent on the characteristics of a specific microprocessor. It is based on the premise that, while processors are continually evolving, the core concepts applicable to all processors do not change. These foundational concepts include the base processor, registers, address space, address space organization, unit of addressability, addressing schemes, assembly vs. machine instructions, encoding/decoding, pseudo instructions, macros, the assembler, linker, compiler, loading and execution. Issues of accessing data, such as addressing modes, symbolic references, relocation issues, dereferencing, data alignment, little vs. big endian, and creation of data instances are other important foundational concepts.

Approach

The approach to the updated microprocessors course at Boise State University is based on three main principles:

- Simplification of the processor
- Making everything visible (nothing is hidden)
- Continual evaluation of results

Simplification of the Processor

Though there is no way to avoid learning some processor-specific characteristics in any microprocessors course, starting with the simplest processor possible minimizes the complexities that can distract from the basic concepts. Simple is more important than powerful for the learning process. Simplicity allows the students to immediately start to explore. There is no sense starting out with the most complex assembly language; rather, it is best to begin with a simple yet elegant platform.

The difficulty with implementing a simple processor approach is that educators' goals are different than vendors' goals when creating processors and development platforms. While vendors do not specifically attempt to complicate their processors and development platforms, this often occurs as a natural result of improving performance and features in order to appeal to industry applications. Modern processors often provide pipelining, cache, and vectored interrupts. Such processors also typically have support for advanced devices such as USB, Ethernet, DMA, etc. If simplicity is the goal, instructors are left with a choice between using antiquated hardware or building their own teaching platform. A solution to this problem is to utilize an FPGA technology-based development board, and to control the processor and peripheral features directly by using a soft core processor. This is the approach utilized for the updated microprocessors course at BSU.

The goal is to introduce the students to an actual processor, but in an educational environment where the instructor has maximum control over the delivery of concepts. A measured presentation focuses on a few core concepts at a time, at a pace that optimizes student comprehension. Unwanted features that distract from the foundational concepts being presented are minimized by simplifying the processor.

Simplifying the processor to eliminate unwanted features involves creating a processor without cache, interrupts, or pipelining; removing advanced devices such as LCD, Ethernet, and USB; and disabling all of the vendor's software support. Many specialized features found in processors will be left out when creating a simple processor for teaching purposes. If students don't need it, they don't see it in the processor. Memory, LEDs, switches, buttons, a seven segment display, and a console device are all that is needed. Use of a soft core processor such as the Altera Nios II used at BSU makes processor simplification easy to do.

For each specific processor configuration, it is possible to create a custom data sheet so the students do not have to spend time wading through information that doesn't apply. Thus the data sheet for the first (minimalist) processor the students interact with would be very small (about 10 pages), and much less intimidating than the 200-300 pages for the full Nios II data sheet.

This simplified minimalist processor is utilized during the teaching of the basics of assembly and the C language (typically the first eight weeks of the course). Once those topics are fully covered, a new processor configuration with cache, hardware interrupts, and other devices is introduced as a processor upgrade. The students participate in implementing the processor upgrade, and learn for the first time that they are dealing with an FPGA. The processors are branded using the seven segment display to indicate which processor configuration is being used.

Making Everything Visible

Vendors will often include software to hide device interactions that may be beyond the understanding of students or developers. Sophisticated software layers are not needed to blink an LED, and in fact can hinder teaching basic concepts of memory mapped devices. The 'nothing hidden' approach involves removing abstractions so that students can see what is happening at the lowest level. Removing the vendor-supplied abstractions allows the students to develop their own simple abstractions and gain experience with this process.

With the removal of the vendor-supplied hardware abstraction layer (HAL), the facilities needed to read or write character data from/to the console (e.g. stdin/stdout) are also disabled. This provides an opportunity for the students to develop their own routines to interact with the console, enhancing their understanding of this process. For example, to print something on the console, the student must create their own print command since there are no functional facilities to print once HAL is disabled.

Both assembly and a subset of the C language are utilized in the updated Microprocessors course at BSU[2]. Students are taught to use the development tools (e.g. objdump) to expose the assembly produced by the C compiler (nothing is hidden). The revelation that C generates predictable assembly instructions enhances students' understanding of the C language and increases their confidence level. It also solidifies their understanding of C pointers because they can see how memory is accessed at the assembly level.

Continual Evaluation of Results

Teaching methods and materials continue to be evaluated for effectiveness and refined where necessary to improve student understanding of the fundamental processor concepts. In addition, the list of basic concepts is constantly under review to adjust the order and pace of presentation (limiting the number of new concepts presented per session), and to re-evaluate which concepts merit inclusion in the category of core foundational concepts.

Trial and Error

As the updated microprocessors course approaches the end of the fifth year since original implementation of the course update, changes continue to be made in both the lecture and lab portions of the Microprocessors course.

Processor Configuration

One significant improvement made at the end of the second year was to permanently preload each processor onto the development board (using a flash configuration device), instead of requiring students to manually configure the FPGA with a processor each time the board is powered on. With this change, the students have no reason to question whether they are using a true microprocessor development board. This simple change dramatically reduced student confusion. The original processor configuration is already preloaded on the board at the beginning of the semester; new processor configurations introduced later in the course are loaded by the students via a scripted procedure as a lab activity.

Course Reference

A standard microprocessors textbook was used for the first year. However, the text did not complement the selected teaching platform or concepts. An in-house course reference with integrated homework has been in development since Spring 2010. Prior to that time, the students were provided with handouts to supplement lectures. Handouts were simply not successful, as the students did not assume any responsibility for the material. Each semester, refinements continue to be made to the reference based on its relevance and effectiveness in the course. Developing and including homework in the reference has made a positive difference, providing a useful means for the students to solidify and measure their understanding of the important concepts.

The current course reference is over 220 pages, and is printed through campus printing services for the students to obtain from the bookstore. There are still several advanced concepts that are currently being addressed in the lecture and handouts rather than in the course reference. Evidence of improved student understanding from using the published course reference highlights the need for developing supporting instructional materials for these advanced topics, and integrating them into the book as well.

Number of Processors

Initially it was thought that the advantage of a soft core processor was the ability to provide a new processor for every lab. It was determined that using the same processor for the first eight weeks provided a better platform for learning the basic foundational concepts. A second processor is introduced with additional advanced devices for the remaining labs, with an optional third processor for the final project.

Constant Refinement

The level of understanding of the core concepts is evaluated by tracking and refining every test and quiz question, based on application of the 70-70 rule (at least 70% of the students scoring 70% or better). If less than 70% of the students receive a score of at least 70% on a particular question, that is considered an indication that changes should be made in how the concept is taught. Some of these changes dictate modifications to the processor - which are readily accomplished with the soft core processor. In many cases, providing additional reference examples and homework have been the solution.

Though removal of HAL was an original goal, it could not immediately be accomplished. Since HAL automatically registers interrupt service routines (ISRs) for devices under its control that require them, it interferes with student-developed ISRs and hampers the learning process. Until HAL could be disabled, the instruction of ISRs was very difficult. Once it was discovered how to deactivate the vendor-supplied HAL support and develop instructional materials for the handling of hardware interrupts, the learning process for writing ISRs significantly improved. This took several semesters to refine.

One of the most important usages of external interrupts and the associated ISR is the external timer device (typically a crystal). A timer device provides the ability to accurately measure elapsed time. With a timer device, many new problems that require accurate time measurement (e.g. pulse width modulation, analog-to-digital and digital-to-analog conversions) can be tackled. With the introduction of ISRs, race conditions were also addressed in order to properly integrate the ISR into the overall solution.

Signs of Success

Comparing grades for the upgraded Microprocessors course over the first three years is not particularly meaningful, since the course was a moving target during that period. However, since the addition of the book and the elimination of HAL, the grade distribution in the course has changed significantly. An analysis of grades in the Microprocessors course (lecture) over the past four semesters reveals an increasing number of students in the passing category.

In the Spring 2010 semester, 70.3% of the students earned passing grades. About half of those were at a C level, which skewed the grade distribution toward the C-D-F end of the curve. By Fall 2011, the percentage of students achieving passing grades had reached 84.8%. More importantly, only a quarter of those were at a C level. (Unfortunately, there are still too many students who “just don’t get it”– typically those who will not do the homework. Even so, the average in the course has increased steadily over the past four semesters, from 72.5% to 77.8%.)

Other signs of success, though less quantifiable, are no less real. Fewer student complaints, increased enthusiasm and improved capabilities have been observed.

With the addition of time measurement, some of the activities that can be done in the lab include the creation of software pulse width modulation (PWM) routines to dim LEDs, vary the speed of DC motors, or produce audible sounds with speakers. The ease with which students are able to accomplish these tasks for final projects has increased significantly each semester, particularly in the past four semesters. Student complaints about the time required in the lab have decreased as they are able to complete the labs in less time.

There have been numerous students with previous C language experience who have indicated that the Microprocessors course was the reason that they finally understand pointers. Another measure of success has been the number of students who announce they have decided they want to specialize in computer engineering.

A noticeable improvement can be seen in elective follow-on courses in the students' abilities to tackle projects with little or no help. It was in one such recent course that a normally quiet student, using a breadboard with a microcontroller, exclaimed "It blinked!" when he finally succeeded in blinking an LED.

Conclusion

Microprocessor or development board vendors will try to sell more hardware by giving free products to schools, and showing them how easy it is to solve problems using their supplied software libraries. While this might be appropriate for courses that utilize microprocessors as a solution, this is not a good approach for studying the microprocessor from an engineering standpoint. The students do not gain skills that they can take to the work place, unless they are fortunate enough to find a job that has identical hardware and software. To empower the students requires teaching them how a processor works, so that they have the skills they need when presented with a different architecture.

Finding a platform that is simple, and not encumbered with too many or advanced high performance features that require work-around processes for beginning students, is a difficult task. If such a processor were found, it would be inflexible as the students' skills grow. There is also no guarantee that the processor will not evolve and will remain available many years into the future.

One very elegant solution is found in the usage of an FPGA development board and a soft core processor. The soft core processor can be placed in flash on the FPGA development board to automatically configure each time the board is powered up. In this configuration, the beginning student is not aware that the board does not have an integrated discrete processor.

Using the soft core processor approach allows the instructor to control the features, and in turn the concepts presented in the class or lab. By isolating the teaching platform from the vendor-supplied development environment, the teaching of a beginning microprocessors class is no longer controlled by what the current industry features provide. If the FPGA-based development boards currently used for the course were to become obsolete, there are currently at least four other off-the-shelf boards that could be used as is or readily adapted to continue using the same processors. Even though the original reason for and usage of the soft core processor in the microprocessors course at Boise State University have evolved, the soft core processor continues to provide a viable and relatively stable platform for teaching the basics of microprocessors.

Bibliography

1. A. Planting and S. Ming Loo, "On the Use of a Soft-Core Processor in Junior Microprocessors Course," 2008 ASEE Annual Conference, Pittsburgh, PA, June 22-25, 2008.
2. S. Ming Loo and A. Planting, "Use of Discrete and Soft Processors in Introductory Microprocessors and Embedded Systems Curriculum," ACM SIGBED Review, Vol. 6, No. 1, January 2009. (http://www.cs.virginia.edu/sigbed/vol6_num1.html)

Join Professor Fernando Reimers in a conversation with seven IEP graduates for a discussion of 'One Student at a Time' in Longfellow Hall (Askwith Hall) from Join Professor Fernando Reimers in a conversation with seven IEP graduates for a discussion of 'One Student at a Time' in Longfellow Hall (Askwith Hall) from Empowering Students to Improve the World in 60 Lessons. 20 September 2017 Â·.Â Join me @FernandoReimers on twitterchat #Leadiep discussing book One Student at a Time. Leading the Global Education Movement on Monday, August 14th from 1-2pm EST Book will be FREE on kindle August 10-11th. Empowering Students to Improve the World in 60 Lessons. 15 July 2017 Â·. What is in this book? Empowering Global Citizens: A World Course. Fernando M. Reimers. 4.7 out of 5 stars 11.Â Another recent book Empowering Global Citizens discusses why global citizenship education, aligned with helping students advance human rights and contribute to the achievement of the Sustainable Development Goals is an imperative of our times. A forthcoming book Empowering Students to Improve the World in Sixty Lessons. Version 1.0, presents a framework and protocols to help teachers and school leaders develop strategies and curriculum for global citizenship education.