Teaching Embedded Systems Design – An International Collaborative Project

Saeid Nooshabadi[1], Jim Garside[2]

Abstract - **This paper covers an international collaborative teaching project involving the design of a state-of-the-art microprocessor and embedded system between the University of New South Wales, Manchester University and the Imperial College, London University. This project, being the first of its kind anywhere in the world, provides an environment that replicates the current industrial practice in embedded system design in an easy and comprehensible setting; an environment where the processor, dedicated coprocessors and software are all integrated to create a functional system such as used in sophisticated electronic devices, including mobile phones, webphones, photocopiers, televisions, digital cameras and PDAs. We see such collaboration as important in both reducing development costs in developing up to date - and increasingly sophisticated - courses as well as fostering good relations between peer departments.**

Index Terms – Embedded Systems Teaching, Microprocessor Design Teaching, Programmable Hardware Development Board, International Collaboration in Teaching.

INTRODUCTION

Increasingly, computer and electrical engineers are involved in the design of computer-based systems to address highly-specialized and specific applications in computer, aerospace, telecommunications, power-production, manufacturing, defense and electronics industries. They design high-tech devices ranging from tiny microelectronic integrated-circuit chips, to powerful systems that utilize those chips and then use efficient telecommunication systems to interconnect those systems. Applications include consumer electronics (CD and DVD players, televisions, stereos, microwaves, gaming devices), advanced microprocessors, peripheral equipment, systems for portable, desktop and client/server computing, and communications devices (cellular phones, pagers, personal digital assistants). It also includes distributed computing environments (local and wide area networks, wireless networks, intranets, internets) and embedded computer systems (such as aircraft, spacecraft and automobile control systems, in which computers are embedded to perform various functions) [1].

We were very concerned with the teaching of microprocessor design in the Australian universities. In general, the practice in the Australian universities is similar to that of technical institutes: to use an 8-bit processor to teach students how to program in assembly language, and to do some simple interfacing to the outside world on a prototyping board using some simple control and measurement experiments. Unfortunately, the practice at the University of New South Wales in teaching embedded systems was similarly very outdated and had not kept pace with the rapid changes in industry practice and expectations.

Modern real-time embedded systems need to perform tasks that are very demanding and require a great degree of sophistication. They need to be able to handle a mixture of data-intensive and control-oriented tasks. They need to have architectural support for multi-tasking, multi-threading, concurrency, polling, interrupts and preemption, as well as user and supervisor modes of operation. Until a few years ago, these features were the domain of desktop PCs and other computers more powerful. In addition, modern real-time embedded systems in most instances are mobile in nature, and hence need to be energy-aware to conserve battery power. They also need to be resource-sensitive to reduce cost. Our belief is that our courses in microprocessor design and interfacing of embedded systems need to reflect such sophistication in their design.

This paper details our efforts to modernize and refurbish the Digital Systems Laboratory, a facility shared between the Schools of Electrical Engineering and Computer Science and Engineering at the University of New South Wales. This laboratory is used by up to 600 students every year.

GRADUATE ATTRIBUTES

Our guiding principles in this project of modernizing our Digital System Laboratory were to emphasize professional practice and engineering design, problem solving and critical thinking skills, communication skills, and teamwork, through the design and execution of well-structured and well-thought-out courses; where theory and laboratory experiences are seamlessly integrated in tune with the latest industry practices. In particular, the outcomes that we wanted our students to achieve as part of their graduate attributes were:

- To possess the ability to design computers and computer-based systems that include both hardware and software to solve novel engineering problems, subject to trade-offs involving a set of competing goals and constraints.
- To be able to use their breadth of knowledge in mathematics and engineering sciences that are associated

[1] Saeid Nooshabadi, School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, 2052, Australia, saeid@unsw.edu.au
[2] Jim Garside, School of Computer Science, Manchester University, UK, jdg@cs.man.ac.uk

with the broader scope of engineering and apply it to the narrow field of computer-based systems design.

- To have the ability to understand the issues, possibilities and limitations of the mapping and representation of a mathematical algorithm associated with an engineering theory into a format suitable for computer-based systems.
- To acquire and maintain a preparedness for professional practice in engineering.
- To be able to design or redesign a product, a process or a system to meet desired needs.
- To have the ability to deal with the essential engineering and computing details, while always being able to keep the "big picture" in mind.

To achieve these outcomes for our students, we set out to:

- Create an environment for the systematic and creative application of scientific and mathematical principles to the efficient design and operation of computer-based structures, processes and systems.
- Create a setting where students can identify deficiencies or weaknesses in an existing solution and try novel ideas to improve it.
- Create an environment where the task of design is fundamental and central.
- Prepare educational materials that have a great deal of content, while at same time teaching students to think and discover for themselves.
- Create an environment in which students will feel that the particular course they have taken is the best course ever.

MODERNIZATION OF THE DIGITAL SYSTEMS LABORATORY, UNSW

A. The Search for a Suitable Platform

In the final quarter of 2001, the development team set out to design a course for the University of New South Wales that was based on modern industry practices. Our path led us to an educational liaison with ARM Ltd [2], the designers of the ARM processor core. Although ARM did not become involved directly in our project, this liaison led to an ongoing collaboration between the University of New South Wales, the University of Manchester and Imperial College, London University. The teams at the University of New South Wales and Manchester University took a proactive role, while the Imperial College took a supportive role.

Since existing commercial development boards were demonstrably inadequate and far too expensive for our intentions, we decided to collaboratively design and manufacture an ARM-based embedded development board, augmented with Xilinx programmable digital logic devices [3] for dedicated data processing, and extended with an auxiliary extension board for easy future development and expansion. The ARM processor was chosen, as it is by far the most popular embedded processor at present time. In fact, due to its low power, relatively high performance, and versatile instruction set, over 75% of the world's sophisticated electronic devices using a 32-bit microcontroller are based on the ARM processor core, including such devices as mobile phones, PDAs, MP3 players, digital cameras, Bluetooth devices, webphones, wireless equipment and so on.

One of the main reasons we chose the ARM processor is that we believe it is very important for students to be exposed to a state-of-the-art hardware and software development platform that is based on a processor used in the actual industry.

A programmable hardware development board was chosen instead of mere simulation in software as this allows a student to see how things work "in the real world". In particular, learning the ins and outs of a hardware platform at the lower level gives students an appreciation of computer architecture that just programming in high-level languages does not. Understanding the mapping between the higher level language and the hardware, and its associated resource implications for embedded systems, is quite a fundamental issue. Learning how some of the higher level concepts, such as operating systems, tasks, scheduling and preemption, actually relate to the underlying architecture support and instruction set is highly important for any embedded system designer.

Major reason for developing our own was that hardware/software codesign is an important issue in computer engineering, perhaps particularly in embedded systems. For example in a mobile 'phone it is important to provide a general purpose processor for `housekeeping' functions. This may not be sufficiently capable to provide all the real-time DSP functions, especially on a limited power budget, hence application specific hardware is also needed. FPGAs now allow students access to a significant uncommitted hardware resource in the same way that microprocessors revolutionised software teaching. Therefore, with a processor and FPGA combination experiments can be made in the trade-offs of partitioning an algorithm at the hardware-software boundary.

One additional aim in designing our own development board was to achieve economies of scale by increasing the size of a production run. All three universities agreed to use the same platform. Another goal of this project was to share the design ideas, laboratory manuals, software and support materials.

The project took about 21 months before it was ready for its first classroom trial for the Microprocessors and Interfacing course at the School of Electrical Engineering, University of New South Wales. Because of its long association with the research and development of the ARM processor, Manchester University had had its first trial earlier. At this time, Imperial College is still at the stage of implementing the new platform. During the course of this project well over 750 emails were exchanged between the teams at UNSW and Manchester University.

B. Development of the Hardware Platform

We went through several stages of rigorous consultation regarding the desired features of the board, the choice of processor, FPGA chips, and other peripheral devices. At one stage, the UNSW team even contemplated designing and fabricating their own board. However, eventually it was decided to join hands with the other two institutions and pool our resources to design a common platform. In a development of such magnitude, where the design and production costs are high, cross-institutional collaboration becomes imperative. In this joint effort, nothing was left to chance, as much as was possible. Every aspect of the design was carefully scrutinised, checked, and rechecked by the UNSW team and the team at Manchester University, until we were all satisfied that it would work. The actual PCB layout of the development board was conducted at Manchester University.

In parallel, the UNSW team designed and manufactured the extension board hardware to use in conjunction with the main development board. The extension board protects the expensive main board from students' accidental or intentional misuse in the laboratory environment. It also allows easy expansion for the future development. During the peak of populating the manufactured boards with components, four technicians were assigned to work on this task full-time for three weeks.

One of the challenges of manufacturing the development board hardware was the extensive lead-time involved in procuring all of the electronic components. This lead-time required us to order these components months in advance, which in turn required us to finalize the design of the development board quite early on.

The overall cost of the board was estimated, and each of the three universities was responsible for obtaining the more expensive items from the bill of materials and sending them to the manufacturer in Wrexham, UK. The smaller items were left for the manufacturer to supply. Cost of manufacturing was shared proportionately between the three establishments.

One big-budget item was the Xilinx Virtex-E FPGA, with a price tag of over $1500 per chip. We at UNSW, and our counterparts at the Imperial College, simultaneously applied to Xilinx Inc. for donations of these chips. Because of our long-standing relationship with the company, Xilinx Inc granted us our request.

A systems-level view of the complete hardware development platform is shown in Figure 1. The DSLMU Microcontroller Board, as we called it, consists of the two boards mentioned above, connected to each other. Figure 2 is a picture of DSLMU Board. The main features of the DSLMU Microcontroller Board are:

- **Main Processor**: The MU Microcontroller Board is based on an ARM7 processor core from Atmel. The particular microcontroller used on this board is the Atmel AT91R40008 device.
- **System Memory:** Connected to the Microcontroller Bus is the system memory in the form of 4 MB of Static RAM and 2 MB of Flash ROM.
- **Field Programmable Gate Arrays:** The main peripherals connected to the Microcontroller Bus are the two Field Programmable Gate Arrays (FPGAs): the Xilinx Virtex-E and the Xilinx Spartan-XL. These FPGAs are designed to be reprogrammed to become any sort of peripheral coprocessors that may be needed.
- **Connected Peripherals:** The DSLMU Microcontroller Board has a number of peripherals connected to the microcontroller, as shown in the system block diagram. These include the serial ports, the LEDs on the MU Board, the LCD module and the Ethernet controller.
- **Uncommitted Peripherals:** The Expansion Board contains a number of uncommitted switches and LEDs. These are "uncommitted" in the sense that they are not connected to anything: students can use these peripherals in their experiments, either connecting them to the FPGA input/output pins or to their own circuits.

Details of the board including the schematics and reference manuals are publicly available at URL *http://dsl.ee.unsw.edu.au* [4].

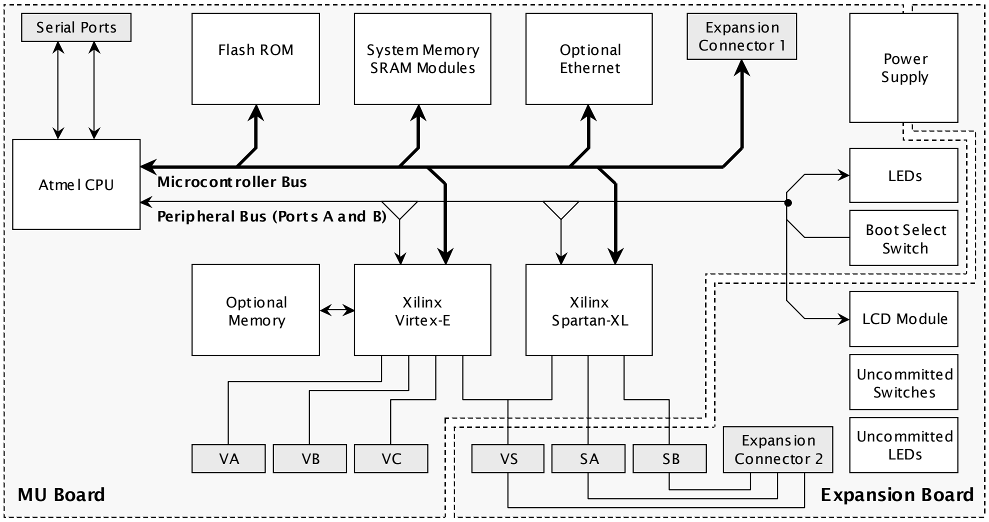

FIGURE 1
SYSTEM BLOCK DIAGRAM OF THE DSLMU MICROCONTROLLER BOARD.

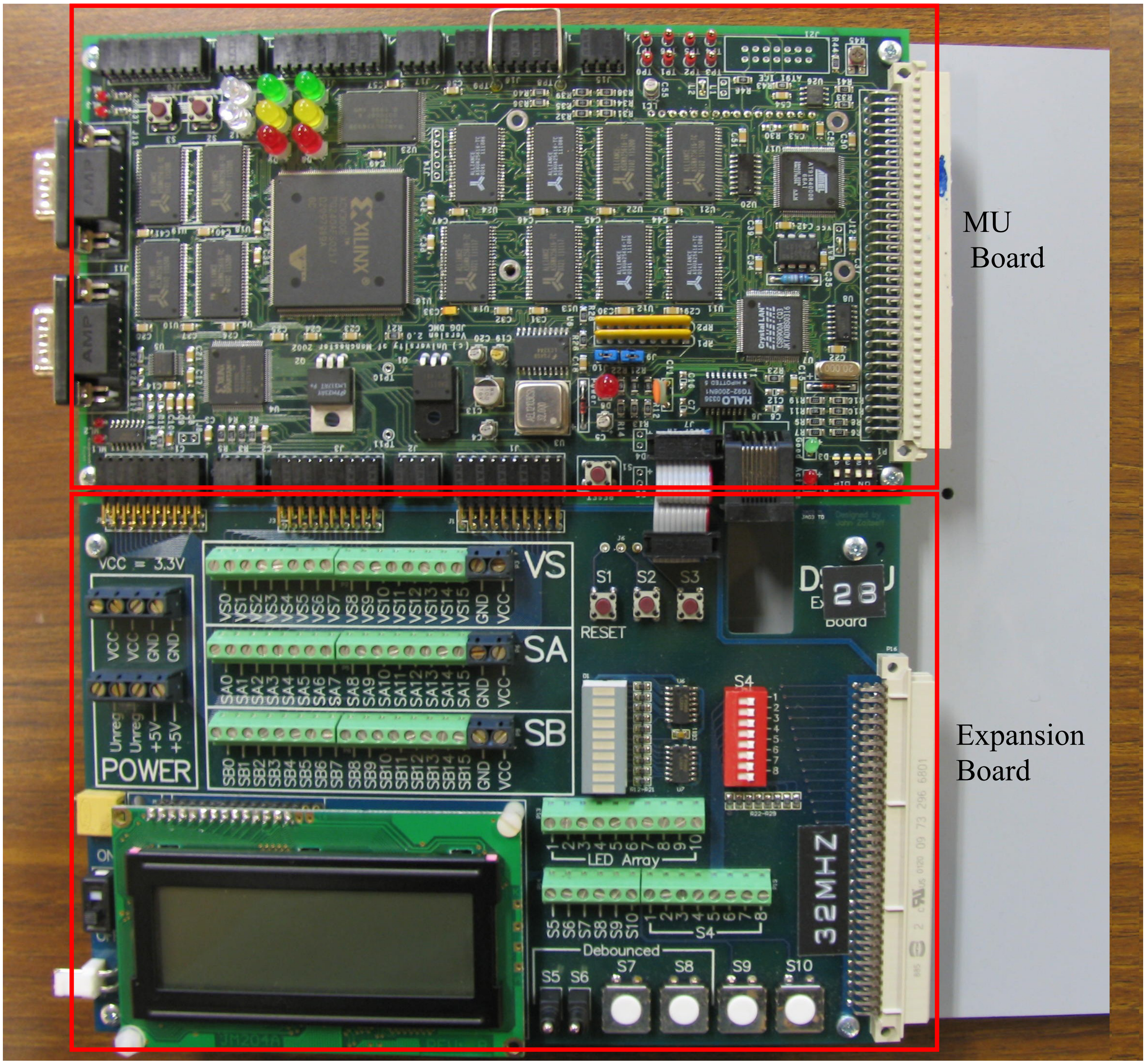

FIGURE 2
DSLMU BOARD WITH MU BOARD AT THE BOTTOM AND THE EXPANSION BOARD AT THE TOP

C. Development of the Laboratory Manual and Lecture Materials

Simultaneously with choosing the software tool-set, we needed to develop the laboratory manual and documentation to be ready for student use for Session 1, 2003. Using a simple commercial board and a bare-bones prototype board that the Manchester University team had put together, the UNSW team designed a set of laboratory experiments for every week of the semester beginning from Week 2. The Imperial College team was also in the process of designing a new course. At that time, Manchester University was one step ahead of UNSW in developing a laboratory manual. However, the UNSW team decided not to use their material for two reasons. Firstly, the curricula and program structures at the three institutions were very different. In particular, the courses at the Imperial College and Manchester University were fully laboratory-based and depended on prerequisite theoretical material from previous courses. The course at UNSW, on the other hand, was the first course in the curriculum area and therefore had substantial lecture content in addition to a laboratory component. The second reason was that we wanted to add to the pool of material developed across the three institutions.

Alongside a set of lecture slides and notes, tutorial sets and weekly on-line quizzes were designed to support the practical experiments. The publicly-accessible classroom delivery materials are available at the URL *http://subjects.ee.unsw.edu.au/~elec2041* [5].

D. Development of the Companion CD-ROM

As all aspects of this development were coming together, the UNSW team came up with the innovative idea of publishing its work on a comprehensive CD-ROM for student use outside the laboratory. From that moment on, the focus of all development activities was to provide an environment where every aspect of the laboratory platform—even the Linux desktop—could be replicated using a few simple steps on the students' home computer. That CD-ROM became known as the Digital Systems Laboratory Companion CD-ROM.

The Companion CD-ROM includes everything the student would need to work from home, apart from the actual hardware, of course. Since the earlier experiments deal with a software simulator only, lack of hardware is not too big an issue. Even for the later experiments, the emulation feature provided by the software and the documentation on the CD-ROM means that the student does not need to waste time in the laboratory writing his or her source code. A set of scripts that students could use to quickly self-check the functional correctness of their program was designed and placed on the CD-ROM. The laboratory exercises were also well written and were quite clear: students knew exactly what was required of them at any time. We subsequently released an on-line version of the CD-ROM to the public at the URL http://dsl.ee.unsw.edu.au [4]. A similar support website was developed at Manchester University around the same time, available at the URL http://www.cs.man.ac.uk/teaching/electronics/ARM_board/ [6].

E. Creating the Right Context

In order to provide the right context in modern embedded systems design, the School of Electrical Engineering moved our course, ELEC2041 Microprocessors and Interfacing, from the 2nd year to the 3rd year of the Electrical Engineering program; a similar course was already being offered in that time-frame in the Computer Engineering program. This allowed students to take appropriate courses in programming languages and computer systems design that could then be treated as assumed knowledge for our course.

LEARNING AND TEACHING PRACTICES

The issues that were addressed in this project were:

- To expose students to the concept of modern embedded systems design, where the processor, dedicated programmable coprocessors and software are integrated together.
- In "real-world" embedded systems, particularly deeply-embedded systems, students might come across digital signal processors (such as used in voice recognition), dedicated hardware for vector processing, interrupt-driven systems, micro-OSs and device drivers, all at the same time. Therefore, we designed our hardware, and the laboratory exercises, to convey that view of embedded systems.
- To challenge students by requiring them to solve real and difficult problems in embedded systems, within the constraints of time, while still keeping them focused and interested. Unfortunately, there are two conflicting requirements in any engineering experiment: the laboratory experiment must be simple enough for most of the students to be able to complete, while being challenging enough to encourage them to think about it in the first place. In addition to a required set of tasks, each experiment has a set of optional and far more challenging tasks. These tasks build on the previous work and on an understanding of the underlying system. This creates an opportunity for students to have fun as well as to enhance their learning experience.
- To provide the course material within the context of previous and concurrent courses taken. One of the problems with the previous offering of ELEC2041 Microprocessors and Interfacing was that it was offered in isolation with no regards to courses taken before, concurrently or afterwards. The first step in the new offering of this course was to move from it the 2nd year to the 3rd of the Electrical Engineering program. This allowed us to present the material in this course within a framework that is familiar to nearly all students in Electrical and Computer Engineering: that of a high-level programming language. We assumed the C programming language as it is widely taught and is what students learn in the prerequisite course. Furthermore, whenever appropriate, examples are provided of applications using embedded systems in the other disciplines that students are exposed to in the past, present or the future. Material is always presented with regard to their relevance and appropriateness in the industrial and research setting. For example, students are taught that the structure of underlying hardware makes it more efficient to represent (`5 × A`) as its mathematically-equivalent form of (`4 × A + A`). This is often the same problem that our PhD students need to deal with in their designs as well. Another example is how to perform a division operation as a sequence of shifts and subtractions, when the underlying hardware does not allow one to do the division directly. A further example is to illustrate the fact that computers have limited precision by showing that (`A + 1) − A` is not same as (`A − A) + 1`, when the variable `A` is a very large number. In each laboratory exercise or lecture presentation the immediate relevance of what the student learns, and where it can be applied in the latest high-tech product that has just been released onto the market or that is under development, is highlighted. We think doing this has a great impact on boosting students' motivation.

- To provide a flexible learning environment. The Companion CD-ROM, the overall laboratory experience, the on-line lecture material, the weekly quiz, the discussion topics on the WebCT (the UNSW Educational Technology and Flexible Teaching and Learning Tool) all allow students to learn at their own pace, solicit help from each other and have a greatly enjoyable learning experience.

PEER-REVIEW, FEEDBACK AND EVALUATION

The new Microprocessors and Interfacing course and its parallel offering as Microprocessors and Embedded Systems has been in place for three semesters. Between 600 to 650 students have taken the new offering of courses to date. We have conducted both formal and informal evaluations of the course by the students themselves. In addition, we have obtained an evaluation from the course demonstrators and technical staff. We have also conducted international peer reviews of the course. Some of the reviews have been unsolicited and some have been invited. The main aims of the evaluations have been to check the following:

- The relevance of the course materials to the industry,
- The organization of the materials,
- The laboratory documentation and exercises, and their usefulness in the learning process,
- The content and the organization of the CD-ROM and its usefulness,
- The organization of the lecture materials and their appropriateness,
- The best and worst aspects of the course,
- The appropriate choice of tools, and suggestions that would make the course better.

Comments from professionals have been very positive and constructive. One important indicator of the success of this course for us was the increase in the number of students in the Test of Research Summer Program offered by the Faculty of Engineering. There has also been a sharp increase in the number of students wanting to do their project with us: We had the highest number of summer scholarship students in 2004. In addition, scores of students who do their 3rd year and final year projects in other areas approach us for advice regarding computing-infrastructure aspects of their projects.

The UNSW Digital Systems Laboratory website and the course websites are referenced by both ARM's [7] and Xilinx's [8] educational program sites.

Comments from the competent tutors and lab demonstrators have been positive as well.

One of our best tutors who was one of our top students, had worked for Motorola in the area of embedded systems, and came to do his PhD with us, said:

"*The work carried out in the new Microprocessors and Interfacing course is quite similar to the job of a software engineer working in embedded systems. In this kind of job, you're required to write a lot of software to deal with real time events, and drive real devices. This sometimes surprised students, when I would explain that, yes, what they are learning is relevant to the real world! I think the course is quite fun!*"

Students' comments about the new course were very positive. Any negative comments received have the element of human nature in them, involving their dealings with the tutors, lab demonstrators, or the lecturers. Some of the students' comments are:

"*This was a very complete course, I really liked the labs too. I found them very helpful in understanding why everything was the way it was*"

"*It is a very well thought out and planned course.*"

"*Lecture notes were well organized*"

CONCLUSION

In this paper we presented a case study in international collaboration in course design and development for computer-based systems and embedded systems. Such collaboration is important in both reducing development costs in developing up to date—and increasingly sophisticated—courses as well as fostering good relations between peer departments. We are not aware of similar collaborative teaching work of this scale everywhere else in the world.

We believe that some of the most helpful aspects of the refurbished Digital System Laboratory include the state-of-the-art hardware platform, the Companion CD-ROM, clear lab exercises and the facility for self-checking. This, coupled with industry-standard GNU Compiler Tools running under the Linux operating system, gives students an insight into real-world embedded hardware and software

REFERENCES

[1] McGettrick, A.; Theys, M.D.; Soldan, D. L.; Srimani, P.K., "Computer engineering curriculum in the new millennium", IEEE Transactions on Education, Vol. 46, Issue 4,

[2] ARM Ltd. Website: http://www.arm.com/

[3] Xilinx Programmable Logic Website: http://www.xilinx.com

[4] Microprocessor and Interfacing Companion CD-ROM website: http://dsl.ee.unsw.edu.au

[5] ELEC2041: Microprocessor and Interfacing Website: http://subjects.ee.unsw.edu.au/~elec2041

[6] ARM/Xilinx Laboratory Board Website: http://www.cs.man.ac.uk/teaching/electronics/ARM_board/

[7] ARM Educational Resources website: http://www.arm.com/community/academy/resources.html

[8] Xilinx University Program Website: http://www.xilinx.com/univ/xup/course.htm