Experiences of a Summer Workshop in Embedded Systems

Kolin Paul **M Balakrishnan**
Department of Computer Science & Engineering
IIT Delhi, India
email: {mbala,kolin}@cse.iitd.ac.in

1. MOTIVATION

In today's scenario, the teaching of courses in embedded systems has become very important. The ever decreasing feature sizes (keeping in tune with Moore's law) has led to the integration of ever increasing number of transistors in the die. Moore's law also has the additional cascade effect of more devices (or functional units) being integrated in the chip. This clearly has led to the chip becoming more "powerful". These chips when assembled on a PCB now have enough capabilities to build large ("embedded") applications. Apart from this, even single chips offer a lot of processing power. For example, a Virtex II Pro [1] chip contains two embedded PowerPC processors (operating at 400 MHz), 100+ single cycle multipliers, resources to build a memory hierarchy, clock managers and high speed serial transreceivers. Clearly using all these features effectively falls into the embedded systems design paradigm. Today hardware designers are compelled to think of at the level of systems rather than at the level of individual components. Similarly software developers for embedded systems requires proficiency in exploiting available hardware resources and they also may have to guide the other system developers in choosing resources appropriately. In the embedded systems domain, system level design is of crucial importance --- this domain fundamentally depends on how effectively we utilize the available resources in an optimum manner. Although embedded systems design is truly interdisciplinary in nature, for purposes of discussion in this paper, we only look at processor based design.

The other reason which motivates us to embark on teaching courses in embedded systems is the trend in industry. Today a systems developer is in more demand than a specialist hardware designer. Also the availability of embedded development boards and kits has made the teaching of such courses feasible.

In this paper, we try to describe our efforts, in teaching embedded systems and systems design in this scenario. In the next section we try to enumerate what we perceive to be the major constraints in teaching such a course. In section 3, we try to list the key enablers which we believe will help in the success of courses in the embedded systems area. We conducted an experiment in the summer (2006) by organizing a Summer Workshop [2][2]. Section 4 describes the structure of the workshop and the results of the experiment. In section 5, we try to summarize our findings.

2. CONSTRAINTS

The structure of the curriculum in most countries, and in particular India, has often not encouraged interdisciplinary work. This has been influenced by almost watertight compartmentalization of engineering disciplines. For example, in most of the engineering institutions in India the students graduating in Computer Science (CS)/ Electronics and Communication Engineering (ECE)/ Electrical Engineering (EE) disciplines are exclusively trained in their own discipline. Our experience with these students shows that the training given to a CS student rarely enables him/her to go beyond building trivial circuits with a number of discrete components. Similarly, students with an ECE or EE background are often found lacking in basics programming skills. Of course, there has been an effort now to introduce a course in Information Technology which is expected to build bridges across multiple disciplines --- however, in most cases this is at a very nascent state.

In general, there is a huge gap in the teaching of a course in embedded systems --- the subject by its very nature straddles and requires knowledge and skills across various disciplines. At IIT Delhi, we allow our students to work on an embedded systems laboratory. This is a 5 credit laboratory oriented course spread over a period of 4 months where the students work in groups of 4/5 and are expected to develop a system. The focus of the laboratory is to build systems which require cross disciplinary skills from CS, EE and ME (Mechanical Engineering. This course is subscribed to by students from CS/EE and ME. Some of the projects that have been done include

1. RFID based personnel tracking system
2. Theatre Lighting System
3. Campus Cycle System

As is evident, the focus is to enable students to develop interdisciplinary skills. The graduating students from most of our engineering Institutions tend to be very good in their parent disciplines. This is mainly due to the fact that our curriculum is designed to train our students in the core subjects of a particular discipline. However, the world is increasingly becoming interdisciplinary in nature as is evidenced by the growing interest in programs like bioinformatics, biocomputing, sensor networks and embedded computing. In the next section we try to identify a few of the key enablers which have helped us to develop a course in embedded systems.

3. ENABLERS

In the last few years, a lot of manufacturers have made available low cost hardware development boards. Most of these have a small FPGA along with a host of peripherals. The advent of high capacity devices from the two major FPGA vendors has been largely responsible for the growth of interest in developing complete applications around FPGAs. These platform FPGAs have large amounts of RAM (both block RAM as well as distributed RAM). These boards also come with Software Development Packages as well as have some development environment for hardware development.

At IIT Delhi, we have in the past, designed and developed an educational board for Co-Design called INCODE. This has been effectively used in the curriculum of this Institute as well as other educational Institutions for teaching courses in digital systems design. The system based on Xilinx 4010 series FPGAs and 8051 microcontrollers allows students to do

a) Hardware development using both schematics as well as HDL
b) Microcontroller + accelerator to implement codesign projects
c) Build complete systems using keyboard, LED, LCD macros

The availability of low cost development boards which have Spartan3 or Virtex FPGAs from Digilent Inc [3] has encouraged people to develop courses ([5][6]) around these boards. The Xilinx University Program has encouraged the use of these boards by providing technical help as also helping many Institutes with donations. These are ideal in developing small and medium size hardware (and many cases embedded systems) projects. The company also makes available the complete synthesis package freely available off the web (WebPACK). The development of systems which have both hardware and software components has been facilitated by the availability of system development tools like EDK and Quartus SOPC [4]. These tools help the student to quickly use a processor (NIOS/MicroBlaze/PowerPC) along with some custom IPs attached to the bus to build solutions. As a case study, we asked students to port baseline JPEG to MicroBlaze and then build a system which had the DCT component running in hardware. A C implementation for baseline JPEG was provided. An implementation of the DCT core was also provided. It took a few days of effort to get the system running on the Digilent boards.

The availability of well documented and properly constructed Board Support Packages (BSP) helped in a fast ramp up as the students did not have to struggle through many of the often annoying (and frustrating) details of understanding all pin connections as well as getting UCFs correct. The availability of such tools well integrated into the development framework (EDK) helps to focus students with diverse skill sets to work cohesively --- in a group of two, we found that one student would look at the hardware details whereas the other concentrated on building systems software for the application. Therefore, the key enablers for developing short courses in embedded systems or hardware software codesign are

a) Boards with a lot of peripheral support and a reasonably sized FPGA
b) BSP package which abstracts away the details of the board
c) Good Integrated development environments which allow software and hardware development and simulation to be performed within the same environment.

Today we do see a lot of development in this area in the open source community. This has helped in the effective dissemination of knowledge as many “amateur” developers have joined the bandwagon to develop IP “cores”. The availability of such cores (some of them are clearly sub optimal designs) from www.opencores.org and such sites helps maintain and grow the interest in platform based design. Once we were satisfied that we do have the basics in place, we decided to offer a summer workshop in embedded systems for undergraduate students based on these new platform FPGAs. In the next section, we outline the schedule of the summer workshop that we conducted earlier in the year.

4. TRAINING SCHEDULE

The engineering curriculum in India requires that all students undergo a compulsory internship program (typically of 50 to 60 days) at the end of their 3rd year of the four year engineering degree program. A sponsor’s help was enlisted and summer workshop program was announced. This internship program in Digital Systems design was run successfully in the previous year also. The summer workshop (SWDSD2006) was planned as a 45 day program. The schedule was formulated keeping in mind the fact that the students had completed 3 out of 4 years of their engineering program. Also the selected students were from different backgrounds --- we picked 6 CS students, 8 ECE/EE students and 2 students who were doing their degree in IT. This ensured that the program had the necessary interdisciplinary flavor. The students were asked to work in groups of two.

The workshop was structured in such a manner that the students were exposed to the hardware and software platforms and crucial design aspects in the first two weeks of the 6 week program. There were lecture sessions on digital design techniques, VHDL Programming, Hardware Software Partitioning and behavioral synthesis. They were complemented with detailed lecture and lab sessions on modern FPGAs and development boards on which they would work. Initially the students worked on the INCODE platform --- this exposed the students to digital hardware design as well as familiarized them with the lab environment. The lectures were primarily held in the mornings and the afternoons were left exclusively for hands on experiments. The focus of the experiments was on system design and correct datapath /control partitioning. It also introduced them to debugging hardware. The INCODE environment as mentioned earlier, allows the development of complete systems including integration of IO devices. Among the experiments that the students did was a GCD Computation – the input was taken from the keypad and the output was displayed in the LED or LCD displays. All the components were available on the development board. This portion of the workshop was, in our opinion, crucial as this helped students build up the necessary confidence to work with hardware as also tinker with the environment to sort out problems like applying service packs, assembling and checking null modems and other cables etc.

We decided that we would like to expose the students to these platform FPGAs and ask them to develop systems based on a microprocessor (available as soft core or hard core) and some IPs. The idea was to get the students to appreciate systems building using some of the features available on the development board (like SDRAM interface, VGA Controller etc). The lecture sessions introduced the concepts of platform FPGAs along with demos to show how a MicroBlaze or a PowerPC based design is done. The students learnt how to instantiate such cores, attach IPs (like UARTs) to the bus in the Embedded Development Kit (EDK) environment as well simulate the entire system using Modelsim. Synthesis steps to get the design running on the target board were clearly explained in the demos. The availability of well documented examples (of “hello world” type programs) whose design included a soft core (Microblaze), UART, hyper terminal etc was a great help in the ramping process. These tutorials are available in the course pages of many universities as well as in the Xilinx website. The experiments were done on the Digilent Boards which contained a Virtex II Pro FPGA and was made available by Xilinx under their University Program. The students were able to complete their preparatory labs at the end of the third week and were in a position to clearly articulate what “major” project they would undertake. The workshop schedule mandated that the participants complete a project and two best project awards of considerable financial value were at stake. The coordinators of the workshop worked with the students to define the parameters of the projects taking care to ensure that pragmatic design choices were made. We give below short descriptions of 3 out of the 7 projects that were done in the workshop to illustrate the complexity of the work done.

a) **Implementation of 2-Player PAC-MAN Game On FPGA**

The objective of the project was to implement a 2-player Pac-man game on the XUPV2P board, so that part of the game resides in MicroBlaze and part of it is implemented on the CLBs of the FPGA. Pac-man is an age old game, in which a small figure (Pac-man) moves around a maze eating small round pellets and saving itself from monsters rushing towards it. The task was to design the entire game on the XUPV2P board. It was decided to design the game controls on software, i.e. controlled by MicroBlaze, and a Graphics Accelerator on the hardware, i.e. on the CLBs. The human player communicates with the system by way of keystrokes from a standard QWERTY keyboard; the keystroke information is translated into certain data and signals by the MicroBlaze and this information is then transferred to a shared memory (which is part of our custom logic) through the OPB(On-chip Peripheral Bus). After this, the Graphics Accelerator part of our custom logic extracts the information from the shared memory and uses this information to generate proper pixel-information for the VDU. The Graphics Accelerator contains a VGA-Timing module which generates synchronizing and blank signals for the proper display on the VDU. The pixel information is actually passed onto the DAC, which passes them onto the VDU port. The VDU is refreshed multiple times within a second by the Graphics Accelerator. The keystrokes are reflected to the player as congruous movement of his or her Pac-man. The monsters are controlled by the game controller program; their movement is effected by an AI logic that takes as inputs the keystrokes given by the user.

b) **Implementation of 256 point Radix-4 FFT module**

The system designed calculates and displays Fast Fourier Transform of the streaming audio input. The audio input is fed through the MICIN port of the AC'97 of the DIGILENT board and the output can be taken from its VGA port. The displayed output shows the power spectrum of the input signal. Due to presence of noise the output gets slightly distorted. The system used LM4550 (AC'97 standard) chip for taking the sampled inputs. The EDK library (available for the Digilent board) controller for the AC 97 was used. The Microblaze then starts taking audio samples, which is then fed to the FFT module. The pipelined FFT module processes the input to produce the output, which is then fed to the Video RAM by the Microblaze after calculation of its power. The VRAM is BRAM, which is shared by both the Microblaze and the VGA controller. Its locations are mapped to the locations of the monitor.

The FFT module is divided into four stages of almost identical computations. This has helped in achieving pipelining and streaming although the amount of memory used is larger than what was required. In the design standard IPIF interface provided by EDK was removed and a custom interface between OPB and the FFT module was added.

c) **Multiprocessor Emulator**

The objective of this project was to develop an emulator for a multiprocessor system using 4 soft core microblaze processors and running an application on it. These microblazes were attached with a common OPB bus through which they communicated with the peripheral devices attached with the bus. Each microblaze had its local memory in which the application for that processor resides along with its local data. The communication between the microprocessors was done through a shared memory which was also attached to the OPB bus. The shared data was stored in the shared memory. The synchronization of the use of the peripheral by the microprocessors was controlled by a Custom IP called SYNC IP which was attached to the OPB bus. Each microblaze had an ID which was to used to synchronize its access to the shared resources.

5. ANALYSIS

Clearly, the focus of all these projects was on the building of a complete self contained system. We were satisfied with the level achieved by the students keeping in view, the constraints that they faced. All of these projects clearly required more that 3-4 weeks time allotted in the workshop. Despite this primary constraint, all the groups did manage to get some (in many cases, all) aspects of the design working. We believe that the following reasons can be attributed for this apparent success:

- Defining a structure for the workshop which enabled the students to initially work on a system which was simpler and well documented. Seeing their non trivial hardware designs work gave them the confidence to aim higher.
- Availability of good instructional material which allowed a fast ramp up. The

easy access to tutorials as well as complete project zips (with HDL code as well C Code for the application along with the project make files) was paramount in students getting interested and working very hard. These tutorials worked out almost off the box and helped the student to gain in confidence.

- The availability of (inexpensive) hardware (FPGA chips and boards) which were completely supported in EDK. This has often turned out to be a major bottleneck as getting to understand board specifics and getting UCFs right often takes a lot of time. The associated BSPs helped the students in not spending too much time in the understanding of pin connections and other board details and they could work with the abstraction provided by the BSP.
- Availability of drivers (HDL as well as Software) for the hardware modules available on the board was very significant. This can be appreciated from this use case scenario. The base installation of EDK did not have the correct driver for SDRAM attached to the board. After connecting the module in EDK and building the system, incorrect results were obtained. This took 3 crucial days to fix with help from the online community. Once the correct drivers were put in the pcores directory, the application worked smoothly.
- The growing number of newsgroups and online communities interested in hardware and embedded systems development. Many of the initial teething problems faced by the students in using the boards or getting MicroBlaze /PowerPC to behave correctly were resolved with 12-24 hours of posting the problem in the web.
- Availability of design templates which the students could import into their designs.

Availability of low cost development boards, good co design software and a wealth of designs available in the web are the salient drivers which could make the students do this much in the limited time span.

6. Conclusion

The experience with the summer workshop shows that in a period of six weeks, it is possible to introduce processor based design on Platform FPGAs to a mixed audience of CS and EE students. The structure requires introductory design classes, ramp up with simpler designs followed with projects involving platform FPGAs on embedded development boards. The use of online material and help is crucial in the process.

7. References

[1] http://www.xilinx.com

[2] http://www.cse.iitd.ernet.in/~kolin/Training/FPGA/Summer%20Workshop%202005.html

[3] http://www.digilentinc.com/

[4] www.altera.com

[5] http://www.eece.unm.edu/xup/index.htm

[6] http://www.cse.cuhk.edu.hk/~phwl/mt/public/archives/old/ceg5010-2005/