EARLY EXPOSURE TO REAL DESIGN AND IMPLEMENTATION IN A COMPUTER ENGINEERING COURSE

M.E. Clarke and J.D. Garside

Department of Computer Science, University of Manchester
Oxford Road, Manchester, M13 9PL, U.K.

mclarke@cs.man.ac.uk

ABSTRACT

Real hardware design comes as a shock to many computer engineering students who are accustomed to closed form exercises which can only be either 'right' or 'wrong'. Unless students experience open-ended design problems leading to implementation as working circuits they will not suffer many of the lessons of real design or appreciate the satisfaction that comes from creating their own circuits in silicon. This paper describes the introduction of such exercises in the first two years of a degree course.

I - INTRODUCTION

In the past the time taken to assemble hardware circuits made it difficult to produce anything but the simplest designs in first and second year undergraduate courses. Recently Computer Aided Design (CAD) tools, Field Programmable Gate Arrays (FPGAs) and educational routes to silicon have allowed the transformation of practical laboratories so that students are producing working designs after a few weeks, complex machines by the end of their first year and standard cell integrated circuits for fabrication in their second year. In this paper we describe a sequence of design experiments which introduces students gradually to complex circuits and increasingly sophisticated design methodologies, enabling them to develop both skills and confidence in digital systems design.

Although engineering is usually classified as a 'science' subject real engineering design requires inspiration and imagination as well as a sound technical background. The teaching of the technical background is relatively straightforward and traditional methods of imparting facts - such as lectures - can prove quite successful. However when it comes to design few people would dispute that there is any substitute for experience; one must attempt to apply the techniques learnt and suffer the consequences.

In practice it is difficult to provide open-ended design-and-make experience for large introductory classes; students usually have to wait until their final year before undertaking such exercises. There are many problems: limited laboratory time, lack of resources and the difficulty of providing assistance and feedback when there is no single correct answer. With traditional teaching methods students receive a very dry and academic view of electronic design and do not experience either the problems or satisfaction of carrying out design for themselves; they become bored with the subject and, in a computer science department, go on to select software options where they find more scope for creative experimentation and have the gratification of seeing the results of their programming efforts. Thus a major objective in the redesign of our laboratories was to provide an interesting and exciting learning experience which would encourage students to pursue specialised engineering subjects to an advanced level. By the introduction of real design exercises in the first two years we also aimed to develop student design skills and to add meaning and understanding to the taught courses.

For the first year course a set of practical exercises has been devised. Each exercise builds on previous work and the amount of direct help, in the form of rigid specifications and documentation, is gradually reduced. Problems are presented which have a multitude of possible solutions and 'hints' (which may be used or ignored as required) are given suggesting different approaches. Guidance is then given to students individually, if needed, to help them with a solution of their own devising. At the same time an increasing number of CAD tools are introduced so that the student will have a choice in design methodology and will be able to appreciate the appropriate tool for each job. Details of this laboratory are given in Section II below.

In the second year there are separate courses on digital systems design and on VLSI systems design. A practical standard cell chip design exercise is already an integral part of the VLSI design course and new laboratory exercises are being introduced this year for the digital design course. The VLSI design laboratory is described in Section III. It consists of the design of a small ASIC through various levels from an original specification and high level behavioural simulation to layout. The objective is to give the students the experience of real ASIC design and to use the same tools and methodologies for a small design of fewer than a thousand gates that would be used for a large VLSI design of tens of thousands of gates.

The sizes of the classes and the number of laboratory hours are summarised in Table 1.

	First Year		Second Year		
	Semester 1	Semester 2	Digital design	RISC µP.s	VLSI design
Number of students	~180	~85	34	67	35
Lab. hours	20	20	16	16	24
Tools used	PowerView/ XACT	PowerView/ XACT	PowerView/ XACT	PowerView/ XACT	Cadence DF2

TABLE 1. Student numbers in various classes

While the tools provided through EUROCHIP have made it possible to introduce realistic design exercises they do not solve most of the problems mentioned above; the development of laboratories to address these problems has taken a lot of time and effort. In Section IV we discuss the problems we met and our experiences.

II - FIRST YEAR DIGITAL LOGIC DESIGN LABORATORY

Students arrive in the first year with widely different abilities, interests and experience of exposure to real design work. The majority have a very limited electronics background and therefore no prior knowledge is assumed. The initial exercise therefore introduces basic gates (AND, OR, XOR etc.) and from these creates some simple circuits such as a single bit full adder and an SR flip-flop. These are constructed on a specially created patchboard which allows the wiring of gates using interconnection leads fitted with 2mm plugs. The states of all the gate inputs and outputs are indicated at all times with small LEDs which makes debugging easy. This level of abstraction is the closest encounter the students have with physical wiring in their first year; this is a deliberate policy to avoid practical difficulties such as defective equipment or missing power supply connections and concentrate on the *logical* construction of circuits.

In the second session CAD tools are introduced and all further practical work is done 'on-screen'. The chosen CAD package is PowerView which is readily (and cheaply) available and is relatively easy to learn. For simplicity the software set-up is such that only the required tools (schematic capture, simulator and waveform display) are easily available. Written walk-throughs are provided for these tools. It has been found that most students can become quite conversant with this software within a few (3-4) supervised laboratory hours. The actual designs required at this stage are simple: a full adder (already familiar) and a multiplexer. Both of these are used as examples to demonstrate the ease of producing multi-bit circuits by the use of instantiation and design hierarchy and both circuits are used as components later in the course.

Simulation is also introduced here, although this is kept simple and the only method explained involves setting up voltage levels by hand on the schematic's inputs and observing the changes displayed here, much like the LEDs on the patchboard. Waveform display is also covered although with such simple combinatorial circuits it is not yet particularly useful.

The remaining sessions in the first semester are used to illustrate some general techniques used in logic design. These are divided into a combinatorial logic problem (a seven-segment display decoder) and a sequential logic problem (a decade counter). These are used to demonstrate the use of Karnaugh maps, state diagrams and state transition tables, each building on the previous exercise. These designs are compiled using the Xilinx "XACT" automatic place-and-route facility and downloaded onto Xilinx gate arrays. The intent is to allow the student to produce moderately large designs without the need for consideration of implementation detail.

The choice of the practical exercises was influenced by motivational considerations. It was deemed important that a *visibly* working circuit was produced as soon as possible to give a tangible feel to the laboratory. Thus the seven segment decoder allows lights to appear which are clearly correct (or not) whilst the counter yields a circuit which runs autonomously and is observable using the seven-segment decoder. This both rewards the student and simplifies the checking process for the laboratory demonstrators.

The second semester in the first year shifts the emphasis onto design. The students are given two exercises with (mostly) complete specifications and some suggestions as to possible solutions. They are then free to implement the designs in their chosen manner, with teaching support available as required. The particular designs chosen are a "Hi-Lo" guessing game and a simple calculator, which allow the gradual introduction of techniques such as the separation of data and control and finite state machines. Both of these exercises use the cells prepared earlier as components in medium scale designs. The exercises are fairly open ended and allow students some implementational freedom, an important motivational factor. Again implementation is provided quickly and simply using automatic compilation and a Xilinx gate array.

II.i - Laboratory Manuals

Both semester's laboratories in the first year are supported by substantial manuals. These contain descriptions of the experiment requirements but are interspersed with chapters of pertinent teaching and reference material. This is complementary to the accompanying lecture course and is intended to be read and assimilated without being overtly presented as course material.

For example notes explaining about buses and Register Transfer Level (RTL) design are included at the start of the second semester as they become required, and state machine design is described - including a worked example - just before the students are required to attempt it. Other notes describe the PowerView tools in increasing detail so that more facilities are 'available' (such as the use of buses and the construction and use of simulation input files). It is left up to the students to discover which of these are suited to their own needs.

II.ii - The Xilinx Gate-Array Board

The Xilinx gate array board used for these laboratories was designed in-house. Its chief component is a XC3030 FPGA which comprises 100 Configurable Logic Blocks (CLBs) and is capable of integrating 3000 equivalent gates[1]. This is supported by a set of key switches, an LED bargraph and twin seven-segment display; the last of these is supported on a multiplexed bus so that Time Domain Multiplexing (TDM) is necessary if two different digits are to be displayed (Figure 1).

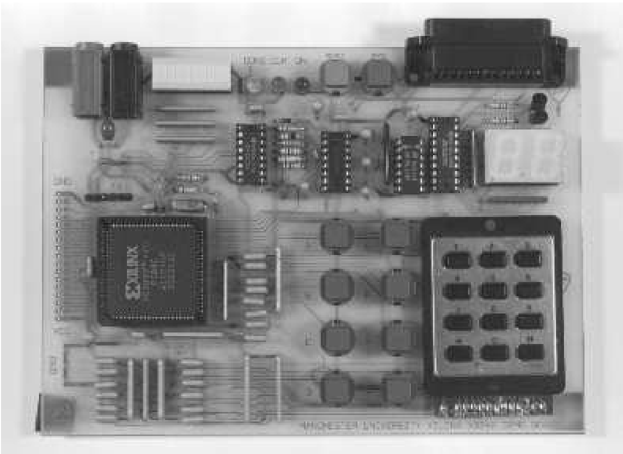

FIGURE 1. The first year FPGA circuit board

Part of the newly redesigned second year laboratory follows up on this facility with a new circuit board. This contains a XC3064 (with over twice the gate capacity of the XC3030) and includes external memory and analogue I/O capabilities. This allows the continuation of the digital design laboratories into the second year, including the construction of simple microprocessors; space precludes a full description here.

III - THE VLSI SYSTEMS DESIGN LABORATORY

The VLSI design exercise introduces students to a methodology for the design of systems on silicon. The design is taken through a number of different levels of design abstraction using a series of CAD tools. The gate level design uses logic synthesis whilst place-and-route tools are used for the layout. A selection of successful designs is then combined on a single class chip which is sent for fabrication as an ES2 standard cell chip. Unlike the Xilinx exercises, only a few students will actually see their designs implemented on silicon, but the important thing is that the possibility is there and the class can see working designs produced by other students - this gives the sense of reality which we are seeking to impart. In fact most of the class are delighted to achieve working gate level simulations and to see their design laid out as a chip - the immediate rewards provided by the CAD tools seem to be satisfying enough.

The exercise starts with a preliminary outline specification of a simple push-button game suitable for young children. A detailed specification is written and a high level behavioural simulation, implemented in C or Pascal, is carried out. By using an already familiar programming language students can

1. Manufacturer's figure

make rapid progress in the early stages without the overheads of learning a special Hardware Description Language (HDL).

Following the high level simulation the students proceed directly to RTL design. They are given a limited set of RTL library components for which behavioural descriptions have already been written. A selection of these together with a control block and a multiplexer block are used to implement their game. Behavioural descriptions of the two student-designed blocks are written in Verilog so that the complete RTL design can be simulated. We are more concerned to teach the *use* of a HDL than to teach the language itself and so provide templates of a Finite State Machine (FSM) and a multiplexer block which can be copied closely. While we would have preferred to use VHDL which is taught in other courses in the Department, Verilog has the advantage that it is completely integrated within the Cadence DF2 system and can be used as input to the Cadence synthesis tool; thus the students do not have to become familiar with yet another user interface.

The gate level design is accomplished entirely by library substitution and by synthesis of the student designed blocks. The appearance of a screen-full of wired together gates after synthesis of the Verilog behavioural descriptions is one of the highlights of the course. Finally each pair of students carries out the place-and-route of their design as an individual chip and records various physical parameters, size, numbers of gates, track lengths, capacitances etc.

The complete exercise takes 24 hours of laboratory time with about 50% more time spent in preparation. Some of the strategies used to achieve what we originally thought would be impossible are discussed in the next section. We cannot yet say whether we have accomplished all our objectives, but it is already noticeable in the end of year exams that students have a much sounder understanding of the design process than they used to.

IV - PRACTICAL EXPERIENCES

The aims of the laboratories are ambitious and achieving the desired result requires a lot of careful preparation and teaching effort. Perhaps the biggest challenge has been to develop worthwhile design exercises which can be performed in limited laboratory time. To do this we have adopted a number of strategies.

For example, the time taken to learn to use the CAD tools could, by itself, be prohibitive. This problem is overcome by integrating most of the learning with the design work. The Cadence and PowerView systems are introduced by means of 3 or 4 hours of walk-through exercises; thereafter manuals giving step-by-step instructions for each of the design stages are provided. A basic minimum of methods is taught, although the quicker students tend to learn more by exploring the systems on their own. This minimal teaching of CAD tools is only possible if capable demonstrators are on hand to sort out problems and misunderstandings as they occur. We have also exploited the fact that there are many concepts that students will absorb during the practical work without being overtly taught. Thus accompanying notes in the laboratory manuals replace several hours of 'lecture-taught' material.

Most importantly, we have learnt to keep designs small and simple. The temptation, for both staff and students, to start out with overambitious specifications is enormous and must be resisted if students are not to get bogged down in so much complexity that they cannot finish the exercise. Quite small designs allow a surprising degree of design choice and scope for originality; despite starting with the same specification every design is unique and plagiarism is not a problem.

One lesson we have learnt is that students need session-by-session 'milestones' during longer exercises. These supply feedback and encouragement during the 'paper' stages of design and give a sense of urgency when a long timescale gives a false idea of the time available and results in a slackening of effort.

Limited resources are another common problem in teaching design to large classes. As a computer science department we are well equipped with networked workstations (Sun 4s) but it is nonetheless very expensive to supply enough machines configured for today's memory hungry CAD tools. The first/second year engineering laboratory contains 28 machines. The first year classes are accommodated by running the same laboratory four times and having the students work in pairs for the first half of the course. The VLSI design students also work in pairs, although this is not just to save resources. It is thought desirable to discuss design decisions with a partner and more can be achieved by sharing the detailed design work.

We find that a student:staff ratio of about 6:1 is necessary for much of the time. Ideally all the demonstrators and staff should be familiar with both the CAD tools and the exercises. In practice this is not always possible and only a few personnel have sufficient hands-on experience of the tools to be able to sort out the tangles some students manage to create. Time invested in training staff is paid back in easing the burden during laboratory sessions. In addition to teaching staff, good systems software support is essential.

CAD tools themselves provide a large part of the answer to the problem of giving feedback on individual designs. Simulation allows students to check their own work and is an effective motivator - once students start to see simulation results they become very involved in solving any problems. We do not attempt to assess the quality of the designs in anything but the broadest fashion e.g. comments on good and bad practice. Certainly in the VLSI course it is quite enough for first-time chip designers to achieve a completed design which works - they are in no position to worry about optimization - that will come later.

Finally, a few comments about the CAD tools themselves - our choices of PowerView and Cadence were dictated by the end products: Xilinx FPGAs and ES2 processing. Having made the effort to set up our classes to use these very complex CAD design suites we find the rapid changes in versions very time wasting and discouraging. It is not only a matter of revising the manuals at least once a year but each new version means yet more expense in machine memory and disk up-grades and a new set of bugs - often with no increase in the functionality. Nevertheless we are obliged to up-date to retain service and library support. Surely there is no need to meddle with the design interface every few months - do we really need four different methods of calling the 'zoom' function? We fear that if the trends continue we will not be able to continue to support such 'state-of-the-art' tools.

V - CONCLUSIONS

The overall intention of these laboratories is to take students with no initial knowledge of logic design and cause them to design fully functional, medium complexity circuits. This is done by a gradual progression in the size (and hence difficulty) of the problem with concurrent experience in the use of CAD tools. Thus in their first exercise students are expected to be able to wire a few (<10) gates together, whilst at the end of the first semester design complexities have reached about 100 gates. By the end of their first year students are able to manage designs of over 1000 gates.

In the second year this increase continues to several thousand gates. The CAD base is also increased; thus HDLs, simulation and logic synthesis are introduced giving a wider picture of the process of production of real integrated circuits.

The proof of the success of our approach is observable in the morale of the students attending the courses. More quantifiably this can be measured by examining the numbers of students who register for the hardware engineering course; these figures - though initially small - have been rising steadily over the past few years, and a small but increasing number of students are now tackling chip designs as final year projects. For example the number registered for the second year VLSI course has doubled in the last two years. Although this trend cannot continue indefinitely it is gratifying to note the interest and apparent enjoyment which can be engendered in these traditionally unpopular subjects.