Efficient Teaching of Digital Design with Automated Assessment and Feedback

1Paul W. Nutter, Member, IEEE, 2Vasilis F. Pavlidis, Member, IEEE, and 3Jeffrey Pepper
1Nano Engineering and Storage Technology Group, 2Advanced Processor Technologies Group
School of Computer Science, University of Manchester
\{p.nutter, pavlidis, jeff.pepper\} @manchester.ac.uk

Abstract—A mechanism for the assessment and timely delivery of feedback on laboratory work undertaken with professional computer aided design (CAD) tools is proposed. The process has been applied to an established 1st year undergraduate course unit covering the fundamentals of computer engineering in the School of Computer Science, The University of Manchester. A concise discussion of the structure of the laboratory exercises is presented, followed by an overview of the traditional approach used (up to recently) for assessing submitted laboratory work. A new automated process is presented whereby students submit their work electronically. Professional CAD tools are used to test designs submitted by students to produce an assessment mark and generate detailed and consistent feedback to the students. The automated marking process evaluates designs using either simulation against a golden test bench, or via a logic equivalence test, depending on the type of design being assessed. The result of a student evaluation demonstrating the merits of the new assessment process is also presented.

Index Terms—Digital design course, microelectronics education, automatic marking

I. INTRODUCTION

Advances in technology and the capabilities of computer aided design (CAD) tools have enabled the realization of complex digital systems and the proliferation of digital devices in today’s society. It is essential, consequently, that an undergraduate curriculum is developed such that it is relevant to today’s industrial practices and encapsulates the latest approaches to design, such as the use of hardware description languages (HDLs), such as Verilog [1] and VHDL [2], which have become mainstream in hardware design [3], [4].

To this end, lectures are accompanied by practical exercises in laboratories, so that students are able to apply the knowledge acquired in lectures [5], [6]. In order for laboratories to be effective in enhancing the students’ learning, the exercises must be designed such that when assessed, formative feedback can be provided to the students on their performance. However, how can this be done effectively, particularly in classes with large numbers of students?

In this paper, we present the development of an automated tool for providing timely and consistent feedback on laboratory work. This laboratory forms the practical element of a 1st year introductory course unit in computer engineering. The proposed marking and feedback process is ideally suited for use for a set of laboratory exercises that use professional CAD tools.

This paper is organized as follows. In Section II, we provide a brief overview of the course unit and the laboratory exercises. The previous approach to assessing the laboratory is discussed in Section III, and the issues with this approach are highlighted. Section IV discusses the proposed automatic marking process that has been developed to enable timely and consistent feedback to the students on their work. Section V provides an evaluation of the proposed process following on from feedback received from students taking the course unit. Some conclusions are offered in Section VI.

II. COURSE OVERVIEW

The School of Computer Science at The University of Manchester has a long history of research in the design of computer systems, from the design of the Manchester Small-Scale Experimental Machine (“The Baby”) [7] in the 1940s, to current research looking at a hardware implementation of the human brain [8]. The teaching in the School has reflected this research with a strong emphasis of hardware design and computer architecture in its undergraduate curriculum. The 1st year undergraduate course unit COMP12111: Fundamentals of Computer Engineering provides the basic knowledge and skills required to support higher-level course units in further years. The course unit is a core module for 1st year Computer Science students and has over 150 students enrolled per year.

A. Course Aims

The aim of the course unit is to provide students with a basic understanding of the design and operation of the hardware that underpins computing systems. A major emphasis of the course unit is on practical design work, with lectures and laboratories running in parallel throughout the duration of the course unit.

Material covered in lectures is put into practice in laboratories using leading edge, industry standard, CAD software (the Cadence® Design Framework [9]), and the industry standard HDL Verilog, which is used for modeling and simulating digital systems. The choice of Cadence® and Verilog was deliberate in order to expose the students to state-of-the-art tools. In this manner, students gain practical experience in using industry standard tools, understand the design methodologies that underpin these tools, and can enhance their employability. The practical exercises in the laboratory are designed to allow students to apply theoretical concepts covered in lectures to solve practical engineering problems. In this way the lectures and laboratories support each other, as well as enhancing the students’ learning.

In total there are 22 one-hour lectures (2 one-hour lectures per week) accompanied by 10 two-hour laboratory sessions (1 per week). Table 1 lists the lecture topics and the laboratory
practical exercises in the order they are covered. The course unit is assessed via performance in the laboratory (50%) and formal exam (50%) held at the end of the semester. The breakdown of marks for the laboratory exercises (as a percentage of the complete lab mark) is given in Table I. There are an additional two optional exercises that account for the remaining 10% of the overall laboratory mark.

### Table I. Lecture and Laboratory Structure

<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture Topic</th>
<th>Laboratory Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to course, Boolean logic &amp; logic gates</td>
<td>No scheduled lab</td>
</tr>
<tr>
<td>2</td>
<td>Number systems, arithmetic, introduction to sequential systems</td>
<td>Exercise 1 Simple Logic Design (15%)</td>
</tr>
<tr>
<td>3</td>
<td>Register Transfer Level (RTL) design, sequential systems and finite state machines</td>
<td>Exercise 2 Binary Addition (20%)</td>
</tr>
<tr>
<td>4</td>
<td>More finite state machines, the design process and CAD tools</td>
<td>Exercise 3 Combinatorial Systems Design in Verilog (20%)</td>
</tr>
<tr>
<td>5</td>
<td>Introduction to hardware description languages, Verilog</td>
<td>Exercise 4 Sequential Systems Design in Verilog (20%)</td>
</tr>
<tr>
<td>6</td>
<td>Verilog finite state machine design example</td>
<td>Exercise 5 Processor Design (20%)</td>
</tr>
<tr>
<td>7</td>
<td>Introduction to processors, MU0, and the instruction set</td>
<td>Input/output in computing systems</td>
</tr>
<tr>
<td>8</td>
<td>Datapath and control design of MU0</td>
<td>Catch-up session</td>
</tr>
<tr>
<td>9</td>
<td>Memories in computing systems</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### B. Laboratory Exercises

There are five practical exercises in the laboratory, as listed on Table I. All exercises (apart from exercise 1) use the Cadence® Design Framework for schematic and Verilog entry (Virtuoso® Custom Design Platform IC 6.1) and simulation of designs (Incisive® Verificiation Platform 13.1). The Cadence® tools are running in Scientific Linux 6.2 (Carbon). The students are provided with a detailed laboratory manual that contains background information for each exercise.

Exercises 3 – 5 are designed so that the resulting designs can be synthesized and downloaded to a custom-built experimental board available in the laboratory (see Figure 1), which includes a field programmable gate array (Xilinx Spartan 3 XC3S200) and an ATMEL ARM9 processor (ATMEL AT91SAM9261). An extender board contains several seven-segment displays, colored light emitting diodes (LEDs), and a bar graph that the students can control directly from their designs. In addition, a keyboard is available to allow input interaction. In order to facilitate interaction with the components on the experimental board a custom symbol called “Board” is provided, as illustrated in Figure 2. Input/output pins from “Board” allow connections to be made to components on the experimental board. In addition, “Board” provides a number of clock signals of different frequencies. A short description of the exercises is provided in the following subsections.

#### Exercise 1: Simple Logic Design

The aim of exercise 1 is to introduce the basic building blocks of digital circuits: logic gates and flip-flops. In order to complete the exercise the students are provided with a custom-built patch board. This board offers a limited number of logic gates from which circuits can be constructed by making connections using wires that are available. Switches are used to set the state of input signals, and LEDs are used to illustrate the state of any output signals. The students are required to construct a number of simple designs, including an exclusive-OR gate (without using the one available), half and full adders, and a D-type latch. In some examples the students are required to use Boolean algebra (which they have met in lectures) to recast Boolean expressions to a form that can be implemented using the selection of logic gates available.

#### Exercise 2: Binary Addition

Exercise 2 provides a hands-on introduction to schematic capture and simulation using Cadence®, in addition, it serves to illustrate the role of hierarchy in the design process. The first task guides the students through the process of producing a schematic design of a half-adder. Following on from this, the students must create three design examples: a full-adder, a 4-bit adder, and a 16-bit adder.

Simulation of designs to confirm that they work to the required specification is an essential part of the design process. Consequently, the students are asked to develop their own
Verilog test stimulus in order to fully exercise their designs. All three designs and test stimuli are assessed.

**Exercise 3: Combinatorial Systems Design in Verilog**

The aim of exercise 3 is to introduce the students to the design of combinatorial circuits in Verilog. Here, the design of a seven-segment decoder must be produced as a Verilog module that takes a 4-bit binary coded decimal value and decodes this to an 8-bit value displayed on a conventional seven-segment display (see page 138 of [10]). The students are then required to simulate their design following the process outlined for exercise 2, where they must create their own Verilog test stimulus.

In this exercise the students are expected to demonstrate a working implementation of their seven-segment decoder using the experimental boards in the laboratory. In order to do this they must create a new schematic. This design is synthesized and downloaded to the board in order to demonstrate their design working in practice. The Verilog module, test stimulus, and working implementation are all assessed.

**Exercise 4: Sequential Systems Design in Verilog**

The aim of exercise 4 is to investigate the design of sequential circuits and finite state machines in Verilog. The exercise is composed of two tasks: to design and implement a modulo-10 counter as a finite state machine, and to complete the design of a traffic light controller to simulate the traffic light sequence at a UK crossroad.

**Modulo-10 counter**

The students are required to implement a finite state machine design for a modulo-10 counter. Once completed the students must simulate their design. However, as this is a sequential system that relies on a free running clock, they must produce a Verilog test stimulus that implements a clock and tests the counter to ensure it operates as required.

**Traffic Lights**

The students are provided with an incomplete Verilog design for a finite state machine that controls the traffic light sequence at a crossroads junction. The light sequence follows the standard UK traffic light signal sequence [11]. The design relies on the finite state machine waiting in each state (corresponding to a light sequence) for a defined period, employing the mod-10 counter design from the first task of the exercise. Once the students have completed the design of the traffic light finite state machine they are required to simulate it. They are required to artificially set the counter values in their Verilog stimulus file. The Verilog module, test stimulus, and working implementation for both designs are assessed.

**Exercise 5: Processor Design**

The aim of exercise 5 is to complete a partially completed processor design – MU0. The exercise supports understanding of processor operation as well as providing a practical example of RTL design. MU0 has been used as an undergraduate teaching aid to illustrate the principles of processor design for a number of years at Manchester [12].

The students are provided with an incomplete datapath schematic that is missing three multiplexers and three system registers; these must be implemented. The configuration of the multiplexers is determined by the MU0 operation, which is covered extensively in lectures. In addition, the arithmetic logic unit (ALU) is missing the 16-bit adder (which they have produced earlier). Once completed, MU0 must be simulated using a memory model containing a test program, which tests the MU0 instruction set and is designed to help in debugging the design.

**III. Traditional Approach to Assessment**

In this section, we describe the traditional approach for marking laboratory exercises that was previously adopted for providing summative and formative feedback to the students. For each exercise the students were provided with a pro-forma answer sheet that should be completed as they progress through the exercises, which contains questions that they must answer. In addition, they were required to demonstrate their working designs to laboratory demonstrators (i.e., teaching assistant, typically a postgraduate student) and have their answer sheets signed off.

Completed answer sheets must be submitted by the exercise deadline, along with printed copies of any designs (schematic or Verilog), test stimulus, and waveform traces. Each exercise is then marked individually, against a marking scheme, and the answer sheet is returned to the student with a mark and hand written feedback by the marker(s) (again, usually laboratory demonstrators). There are a number of disadvantages with this approach:

1. The large number of students (150+ per year, in laboratories with a capacity of 42 students) means that there is a considerable marking burden for each laboratory exercise (an average of 750 answers sheets in total). As a result marks and feedback are often returned to the students late, which is of little benefit to them.

2. In order to provide feedback to the students in an adequate time frame, marking is distributed amongst a number of laboratory demonstrators, this results in:
   a. inconsistencies between marks awarded by the different demonstrators, and
   b. inconsistencies between feedback provided by the different demonstrators.

3. The submission process results in the students having to print a number of documents needed for marking (schematics, Verilog behavioral descriptions, test stimuli, simulation waveforms).

The proposed automated approach to marking was developed to address these issues.

**IV. Implementation of Proposed Approach**

In this section, we describe the new automated process for marking laboratory work. The process has only been applied to the submission of exercises 2 – 5, as the nature of exercise 1 makes it difficult to implement an automated submission and marking process. In addition, by retaining the old approach for exercise 1 it gives us a comparable method to judge the success of the proposed approach. In order for the automated process to work effectively, the laboratory was changed so that the students were provided with all the design files they need to complete the laboratory. In these the design interfaces were
clearly defined (input/output pins provided in schematics and module headers provided in Verilog files), so that the interface matched the interfaces of the test benches used for marking.

The new approach consists of three stages:

1. On completion of each exercise the students should submit their work electronically. The time and date of submission is recorded.
2. Once the submission deadline for the exercise has passed the automated marking process is run. The marking script marks the exercises and provides personalized feedback to the students by email.
3. In the following scheduled laboratory the students must arrange a face-to-face marking session with a laboratory demonstrator.

The second stage is the cornerstone of the new assessment method and is described in detail in the following subsections.

A. Automated Marking Process

Once the deadline for an exercise has passed then all designs submitted are marked using a marking script. The process and CAD tools used for marking the design depend on the type of design being marked.

Two different approaches have been adopted for testing the designs submitted by students: simulation (for simple combinatorial circuits and for testing test stimuli) and logic equivalence testing (for sequential systems). Whilst the use of simulation allows a simple means of setting input parameters and measuring the corresponding output, it fails to work correctly in the case where a system has a state. Any errors in such a system are likely to propagate resulting in the students being penalized repeatedly for what may be a single mistake. The use of logic equivalence testing, which is the novelty of our method offers a fairer approach. Using a formal equivalence checking tool, however, can have a computational overhead.

B. Assessment of designs by simulation

If a submitted design is a combinatorial circuit, the automated marking process assesses the design through simulation using the Cadence tools. First the student’s design is instantiated in a Verilog test bench containing a set of test data covering the test strategy for the design. Each set of test data is taken in turn and applied to the design and the result from the student’s design is recorded and compared to the expected result. The expected result is extracted from a Boolean expression or read from a look-up table. If the results match, the mark record for the student is updated according to the marking scheme.

If the results do not match then the feedback is updated to highlight the test data that failed, the result observed, and the expected result. This process is repeated until all of the test data defined in the test bench has been tested. At this stage a mark has been generated and recorded for each student, and feedback is generated including generic text outlining what was being tested and how.

C. Assessment of designs by LEC

To assess the design of a sequential circuit we have adopted the use of logic equivalence testing; the process is illustrated in Figure 7. This approach does not rely on the need to test sets of test data, which is more difficult where state is involved. Instead, the design submitted by the student is compared functionally against a repository of reference designs, some conforming to the design specification, others not. Associated with each of these reference designs is a mark and feedback that have been generated previously. First the student’s design is loaded in to the LEC tool (here Synopsys Formality [13]), then each reference design is loaded into the LEC tool in turn and compared with the student’s design. If the designs match, then the corresponding mark and feedback for the reference design being tested is returned to the student. If the designs do not match, then we move on to the next reference design in the list. If all the reference designs are tested without a match being found, then the design needs to be marked manually. Once marked manually the new design is added to the repository of reference designs for future marking. In this way we continue to construct a library of reference designs.

D. Assessment of test stimulus by simulation

The goal of testing the student’s test stimulus is to see how effective their testing strategy is at checking the functional behavior of their design. Simulation is used to monitor the test stimulus and check for particular data patterns being exercised.

First the submitted stimulus files are incorporated into a Verilog test bench, which monitors the change in state of the variables in the design. When a change in stimulus is observed the simulation checks the state of the variables against a list of expected tests. If the test is not an expected one, then the simulation continues to monitor the variable until they change again. If the state of the variables is expected, then a check is made whether the particular combination has been observed before, if not, then the test record for the student is updated to keep track of the tests performed. After a time the simulation is stopped and the marks for the test stimulus are determined by observing which tests have been performed. Any missing tests are reported back to the student in the feedback.

E. Face-to-face marking session

Following submission of their work the students are required to arrange a face-to-face marking session in their next scheduled laboratory session. The aim of the face-to-face marking session is for the student to demonstrate their working designs for which they are awarded further marks beyond those awarded for testing the “functionality” by the automated marking process. It also offers the students the opportunity to discuss any issues about their work and obtain some formative feedback.

V. Evaluation

An evaluation of the performance of the new laboratory assessment and feedback structure was performed by asking the students to complete an online survey rating of the laboratory. The students were asked to rate how much they agree with a set of statements with a rating scale of: 1: strongly agree, 2: agree, 3: neither agree nor disagree, 4: disagree, 5: strongly disagree.
The results from this survey are presented in Table II. The response rate for this survey was 45% (out of 181 students enrolled for the 2013/14 academic year). As the automated marking system has only been applied for the past two years, the goal of this survey is to determine the strengths and weaknesses of the new approach. In addition, the parts of the marking process that require further improvement are identified.

The lowest mark (i.e., the highest confidence in a statement) is given to the successful link between the course material and the lab exercises. This response is not surprising as the labs emphasize the teaching of the CAD tools and HDL, which cover a considerable portion of the taught material. However, the response indicates that the lab exercises can be further improved to better cover other parts of the course material, such as the Input/Output communication.

Most of the students also considered the combination of automated/faceto-face marking a superior approach. This is a very encouraging outcome for the new scheme and can be attributed largely to the “fairness” that the automated market scheme guarantees. The students are asked to compare the new scheme with the written feedback/marketing received for exercise 1. Although this exercise is relatively simple, marking discrepancies are unavoidable as a large number of demonstrators (usually 4 to 5) mark this exercise. This number of markers is due to the requirement to provide timely feedback such that the students can perform better in the following exercises. Although the students favor the automated process, they also show a preference into receiving face-to-face marking succeeding the automated feedback. Students value this mixed process, as each of them can discuss with a demonstrator any unclear points. In addition, the instructors have the opportunity to ascertain the understanding and the relating difficulties of the students. Questions emerging during face-to-face marking are often discussed in subsequent lectures to improve understanding of the entire student cohort.

Students, on the other hand, partially agree with the statement that the feedback from the automated marking is adequate in pointing out their mistakes. This is currently the greatest challenge we face. Automated marking guarantees fairness but there is limited capabilities to describe in detail the expected input from the students or why the submitted test benches are incomplete, for all possible cases. The automated feedback cannot be as descriptive or detailed as student’s desire. Although our effort is to mitigate this situation through the additional face-to-face marking, there are still margins for improvement. The goal here is to continue to develop the database of responses over subsequent years, for the various mistakes encountered across submissions. This task, however, must be carefully addressed, as there are some chances to increase confusion instead of clarifying the situation.

We should continue to collect data of this nature as we improve the automated feedback to check whether student satisfaction and understanding improve. The removal of marking ambiguities appears as a strong incentive to proceed further with this scheme, where the marking process is appropriately controlled for a large body of students.

VI. CONCLUSION

A new automated approach for marking laboratories in a first year undergraduate course with a large body of students is presented. The approach employs features of professional CAD tools, such as the logic equivalence checker for accelerating marking and eliminating marking discrepancies. Using this approach we found that marking and feedback were consistent to all students, and delivered in a timely fashion in order for the students to benefit from the feedback provided. A recent survey showed that students appreciate the efficiency of the new method. The next step is to further improve the method by tuning the automatic process to offer more formative feedback for each exercise and possible types of errors.

ACKNOWLEDGEMENT

The authors are grateful of Dr. J. Garside, Dr. E. W. Hill, and Dr. L. A. Plana for their advice and suggestions.

REFERENCES


| Table II. Mean & Standard Deviation of Feedback Scores |
|---------------------------------|------------------|
| How much do you agree with the following statement? | Feedback Score |
| | μ | σ |
| “The lab exercises helped me in my understanding of the material taught in lectures.” | 2.1 | 1.2 |
| “I found the feedback provided by the automatic marking useful in helping me understand where I made mistakes in my design/when simulating.” | 2.9 | 1.1 |
| “I found the face-to-face marking in labs invaluable in helping my understanding.” | 2.3 | 1.3 |
| “I preferred the automatic/faceto-face marking of exercises 2-5 compared to the written feedback for exercise 1.” | 2.3 | 1.3 |