

A Computer Based Software for Student Testing of Analog Electronics Circuits

Charles Hacker

Faculty of Engineering and Information Technology,
Griffith University, Gold Coast Campus, Qld, Australia.
C.Hacker@griffith.edu.au

ABSTRACT: The use of self paced computer based competency testing provides a valuable learning exercise for students. This paper describes the development of a computer based competency testing software, called *AnalogTester*, that presents analog circuit theory questions for unsupervised student testing. Competency testing involves students retaking tests, hence the software contains an extensive data base of questions, and furthermore electronic component values in questions are randomly assigned at test time, to ensure different questions with each re-test. This necessitated that the software, at run time, dynamically generate solutions for comparison to the student entered values. Also, to allow for unsupervised student testing, extensive encryption of the software data files and student results files were necessary, to ensure no deception in the student submitted results. This software complements other software modules, collectively called *WinLogiLab*, previously developed for the teaching of digital electronics. Thus the *AnalogTester* software is the starting software module, for an eventual complete analog electronics teaching software package to replicate the authors successful *WinLogiLab* digital teaching software.

INTRODUCTION

Student *self paced competency testing* is used extensively in the undergraduate electronic courses lectured by the author. The term self paced competency testing refers (in the authors case) to the student being able to re-take the tests, at their own convenience, until a suitable grade has been obtained.

Self paced competency testing has been determined by the author [2] to be a valuable learning exercise for students. This was based, firstly, on the better student completion rates. This was proven with both, more students 'staying' with the course to the end, and with students obtaining higher percentage grades in the final examination. Also the student evaluation surveys, provided to students within the course, have strongly indicated the students believed the competency 'Mastery Tests' were beneficial.

Although the Mastery Tests were proven to be beneficial for the student, the initial paper based Mastery Tests were found to be workload intensive for the lecturer. The high workload was due to the amount of marking, and in the time and effort required for setting and supervising tests. This usually necessitated that the lecturer reduce the total number of times a student could re-sit an individual test, and limit the duration allowed for students to complete an individual topic test.

By utilising *Computer Based Mastery Testing*, the workload was significantly reduced. This was due to the computer both setting, performing, and marking the student's test. A further benefit is that the student could undertake the test as many times as they require, they could undertake the exam at their own pace, in private, when it was convenient for the student.

The author thus developed software, called *AnalogTester*, to perform computer based mastery testing in analog electronics. The *AnalogTester* software provides analog circuit theory

questions for unsupervised student self paced testing. For the software to be viable, the software was required to be firstly, 'secure' enough, to allow for unsupervised student testing. Secondly, for Mastery Testing, the software was required to provide enough differing randomised questions, so that a new set of questions were provided to the student with each subsequent re-test. Finally, the software was required to be freely available for students use, for both on and off campus.

EXISTING ANALOG TESTING SOFTWARE

Analog circuit testing software have been available for some time, and the application '*Circuit Tutor*' [3] by Prof. Burks Oakley II is one of the most popular, being adopted by many universities [4,5].

The '*Circuit Tutor*' application has been utilised in the authors analog electronic courses since 1997, and has constitutively been a valued computer based educational resource. The '*Circuit Tutor*' application offers students very informative, step-by-step tutorials on solving analog circuits, as well as student quizzes, and offers full student testing.

The student testing facility of '*Circuit Tutor*' has actually been (in the past) utilised by the author for student testing purposes. However, there has been a number of limitations in the testing facility that had resulted in a dire need for an updated application.

The main limitation in the '*Circuit Tutor*' software, is that the software is very *in-secure*. That is, it is very easy for a student to fake, or modify, the results output. For example, the software's final result is an image designed for printing. An enterprising student could easily copy this image to a computer 'paint' package and modify, or even totally fake, the given information.

Thus when utilising this testing facility, the author was required to strictly supervise the students undertaking the test, to ensure no fraudulent actions took place. However, the need to supervise the students, negated a lot of the benefits for utilising computer based testing. For example, students could not undertake it at their own time, or at their own pace, or privately in their own home, and required lecture supervision time.

The other main disadvantage is that the ‘Circuit Tutor’ application did *not* directly cover the material within the authors analog electronic courses. Similarly, the ‘Circuit Tutor’ application did *not* offer any method for adding or modifying the testing data base. Thus the authors developed *AnalogTester* application, also includes question generation capabilities. The question generator enabled the development of questions crated specifically to the course content, and allowed for continuous modification and updating of the question sets.

Another disadvantage in the ‘Circuit Tutor’ application, is that it provided minimal student result statistics. That is, the ‘Circuit Tutor’ testing output only provided the students name and score, with NO other information. However, the authors developed software records a full set of statistical information with the student’s saved result file. For example, the recorded statistics included, which questions the students passed / failed, exactly which questions (from the data base) were presented to the student, the time the student spent on each question, and for the Mastery section, how many times the student attempted the individual test.

A further disadvantage in the ‘Circuit Tutor’ application, is that (due to its ‘age’) it is not designed for the latest version of MS-Windows. Thus there were difficulties in installing and utilising this software within the security conscious MS-XP student computer laboratories, of the author’s University.

IMPLEMENTING THE ANALOG TESTER SOFTWARE

The author has previously developed a complete software suite, called ‘WinLogiLab’ [1] to provide tutorials and testing in digital electronics. As part of this software suite, a full student testing application, called ‘LogicTester’ [2], has been developed. The previous work on the digital testing software, provided the basis for developing this current ‘AnalogTester’ application.

By utilising the code for the digital testing software, there was already a frame work for developing a fully interactive, user-friendly analog electronic testing application. The existing code provided for such things as, accessing an external (and updateable) question database, for producing randomly assigned student tests, for computer grading of such tests, and had full encryption security to allow for unsupervised testing of students.

However, the biggest change within this *AnalogTester* software, was the improvement in the ‘randomness’ of the student questions. The existing *LogicTester* code relied on presenting a student with a set of ‘different’ questions (with each subsequent test), by randomly selecting a subset of questions from a large data base of set constant questions (and constant answers).

Yet, with the ‘AnalogTester’ software, the code was enhanced to provide for dynamically altering individual questions, as well as providing an extensive data base of questions. That is, for each question, the individual components (such as resistor, capacitor, and inductor values) are randomly assigned on presentation of the question. This meant that although the student may (if they repeat the topic test a number of times) get the same circuit question, the individual values of components in the question will be different due to randomly assigned values.

However, these random dynamically constructed questions meant that there could no longer be set constant answers defined for each of the individual questions. Instead the software is now required to dynamically generate solutions, at run time, for comparison to the student entered values.

The need to dynamically generate solutions meant that a full mathematical interpreter engine was required. The programming language utilised (Borland Delphi) contains an inbuilt design time interpreter. However, since the inbuilt interpreter evaluates at design time only, yet the circuit equations are required to be developed and evaluated during runtime, meant the inbuilt engine could not be utilised.

More so, the required mathematical interpreter needed to have more capability than the standard inbuilt math engine. That is, the math engine needed to handle complex numbers (for many analog AC circuits utilise complex notation), and it needed to be able to solve full variable sized matrices (since many circuits are solved via matrix methods).

A full complex number *Reverse Polish Notation* (RPN) interpreter, and a *Gaussian Elimination* matrix solver was designed and coded for the task.

DESIGNED APPLICATION

The flow diagram of the *AnalogTester* software structure is given in Figure 1. The *AnalogTester* software starts by requesting the user to either start a new User Test file, or to load an existing saved User Test file.

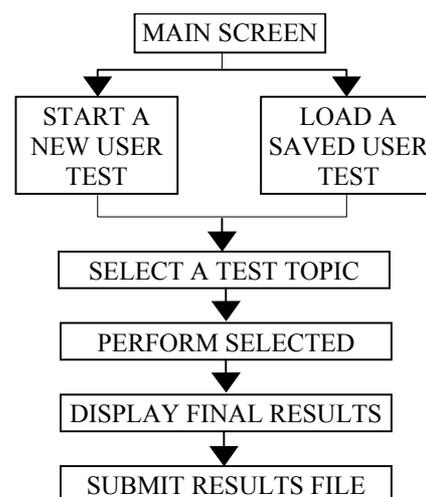


Figure 1: Flow diagram of the *AnalogTester* software

When selecting to start a new User Test file, a screen opens to request for student information. An example New User input screen is given in Figure 2. The requested information desired includes the student name, student number, and a user password. All this information, along with the password, is

stored in the encrypted student Test file. Then, later, upon loading a saved User Test file, the software requests the entering of a password, and will only allow access if the entered password matches the stored password.

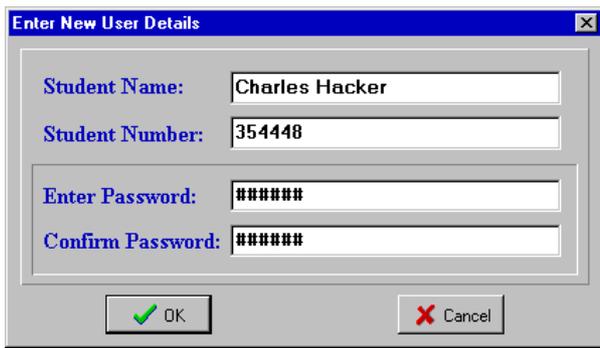


Figure 2: Input a New User Screen

Encrypting the saved User Test file ensures that the file will have no meaning when attempted to be viewed or accessed by auxiliary software. This makes it near impossible for a student to modify the saved file in any intelligible way, (thus preventing a student from falsifying stored results). In addition, the question database is also encrypted. Thus student's are not able to interpret the data base file (thus cheat), if they attempt to 'view' the database by auxiliary software.

Once a new User Test file has been established, or a saved User Test file is loaded, a Select Test Topic screen is displayed, as demonstrated in Figure 3. This screen allows the user to select a test from various topics in analog electronics. These topics approximately follow the weekly lecture topics presented in the author's undergraduate analog electronics course.

Select Test Topic					
Test Summary of: Charles Hacker					354448
Topic	Question Title	Attempted	Correct	Done	Score
1	Basic Circuit Theory	Yes	24	24	100%
2	Circuit Network Theorems	Yes	24	24	100%
3	Single Loop Circuits	Yes	18	18	100%
4	Multiple Loop Circuits	Yes	10	10	100%
5	RLC Filter Circuits	Yes	19	19	100%
6	Transistor Amplifier Circuits	Yes	15	15	100%
7	Power Amplifier Circuits	Yes	12	12	100%
8	Operational Amplifier Circuits	Yes	12	15	80%
9	Analog and Signal Processing	Yes	10	10	100%
10	Oscillators and Active Filters	Yes	11	12	92%
11	Power Supplies and Regulation	Yes	10	10	100%
12	Phase Control and Chopper Circuits	Yes	10	11	91%
13	Communication Circuits	Yes	13	13	100%

Figure 3: Test Topic Selection Screen

As stated, there has been an extensive question database developed for each of the diverse teaching week topics. This allows for completely different sets of questions to be presented, with each subsequent retake of a topic test. Furthermore, as stated, the individual electronic component values are randomly allocated in a given test. This doubly ensures the uniqueness of these particular questions, especially for competency based testing, where students can continuously retake tests.

The Select Test Topic screen also displays the results of previously attempted test topics. This enables students to

monitor their progress, and enables them the opportunity to retake topic tests that have not meet the required percentage score.

Once a test topic has been selected, a question screen is presented, which is based on a circuit diagram. This is demonstrated in Figure 4. The student is requested to answer questions on the circuit, by solving some property of the circuit, or designing some aspect of the circuit.

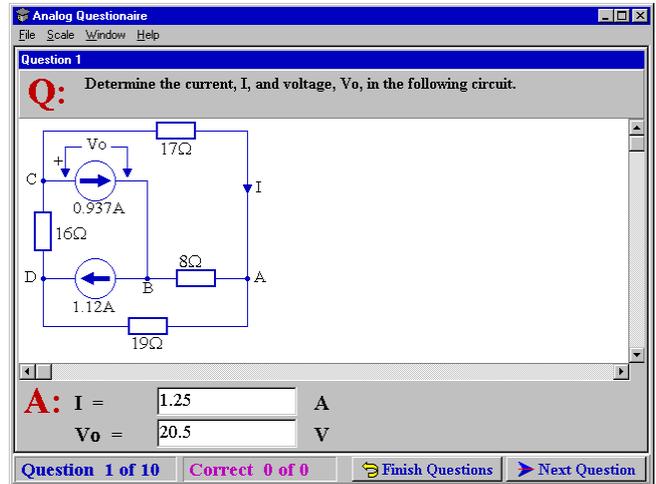


Figure 4: Example Multi-Loop Circuit Question

Once the student has entered their answers, and requested to continue to the next question, the program checks the students answers. The program then displays a *Correct* or *Incorrect* response message, as demonstrated in Figure 5.



Figure 5: Example Result of an individual question

As shown in Figure 5, the response message also lists the answers that were expected by the program. This capability was not initially included in the software, but was strongly requested by the students. Thus the displaying of the expected results apparently provides valuable feedback for the student.

Once the user ends the selected question topic, by answering the required number of questions or manually ending the test, then a final result screen is displayed. An example of this is provided in Figure 6.

The result screen can be copied to an image, or can be printed, to provide a permanent recording of the student results. This recording could be utilised for back-up proof that a student has completed a test. Therefore, to ensure that no deception has occurred, by the altering of the result screen image, a security check string is provided. This string can thus be decoded, and used to verify that the submitted result image is unmodified.

For the analog electronics course lectured by the author, the students are required to periodically submit their encrypted saved Test file, for examination purposes. This test file can then be decrypted by the lecturer, and provides student results, along with many other statistical information.

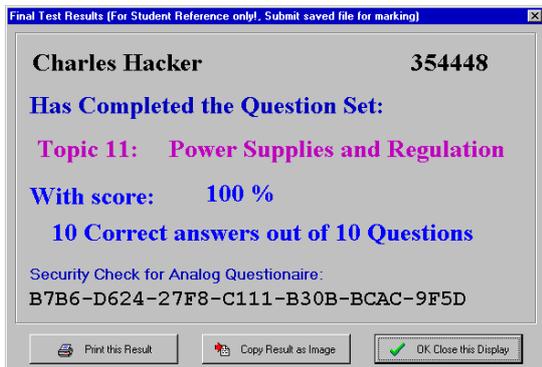


Figure 6: Example Results of Completer Test Topic

The submission of the saved Test file is performed electronically, via Email. An example of the Email submission of the Test file is given in Figure 7. Note, if Email is unavailable, the Test file can be submitted on disk.

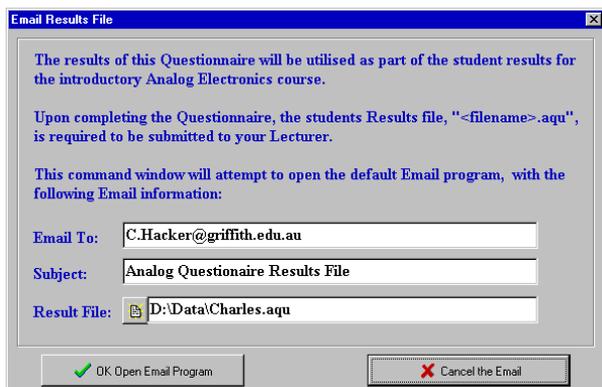


Figure 7: Email Submission of Test Results

TEST BUILDER

Along with the *AnalogTester* module, another *TestBuilder* module was developed, to provide for entering new tests to the data base, and for modifying previously stored tests. An example of the TestBuilder module is given in Figure 8.

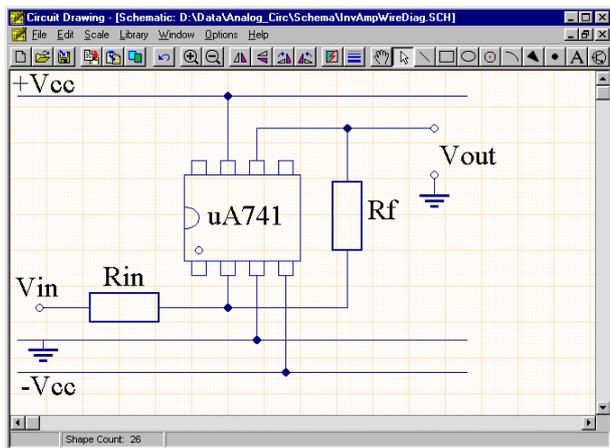


Figure 8: Example circuit entry in TestBuilder

The *TestBuilder* module enables such tasks as, entering of a circuit diagram, entering an equation governing the circuit, and

setting which component values (such as resistor and capacitor) will be randomly defined at run time, and the range for these values.

The *TestBuilder* circuit diagram input section utilises the familiar circuit schematic drawing tools, as with any circuit schematic editor. For example, the drawing tools include lines, circles, curves, arcs, arrows, rectangles, and a library of pre-defined components. These drawing objects are fully editable, resizable, and moveable.

Due to the fact that the *TestBuilder* module provides a convenient schematic diagram generator, this schematic generator capability has been extracted from the module, and is available as a separate 'Schema' application. This 'Schema' application is also provided to students, and can be utilised by students for 'drawing' circuits, when generating laboratory and assignment reports.

CONCLUSION

This paper describes the development of an analog electronics testing software module, called *AnalogTester*. This module provides for secure Competency Based Testing of students. Competency Based Testing has been proven to provide a valuable learning exercise for students. For example a successful test will verify to the student that they have obtained the required knowledge in analog electronic theory and circuit design.

Utilising computer based testing has been shown to reduce the teaching workload in the authors analog electronics course. This was mainly due to the inclusion of encryption and security checks, that allowed the software to be utilised for unsupervised student testing, but still provide valid grading information.

The software is secure, and free for students. Thus the students can install the software on their own computer, or utilise the university 24hr access computing laboratories, to undertake a self test at their own convenience, at their own pace.

REFERENCES

1. Hacker, C. and Sitte, R., 2002, Interactive teaching of elementary digital logic design with WinLogiLab, *IEEE Transactions on Education*, Vol.47, Issue.2, pp. 196-203.
2. Hacker, C., 2003, Computer-based software for testing students in digital logic theory and design', *World Transactions on Engineering Education*, Vol.2, No.2, pp. 281-284.
3. Burks Oakley II, 1993, *Circuit Tutor: Windows Version for IBM/Pcs*, Addison Wesley, ISBN: 0201513706, April 7, 1993.
4. *Downloading Circuit Tutor*, 2001, VCU School of Engineering, Retrieved: November 19, 2004, from <http://www.engineering.vcu.edu/comp/ctutor.html>
5. *ECE220 Instructions for Loading Circuit Tutor*, 1997, The University of Arizona, Retrieved: November 19, 2004, from http://www.ece.arizona.edu/~ece220/Course_Software/Software/ct.html