
A Computer-based Interactive Teaching Software for the Tracing of Logic Levels in a Digital Circuit*

Charles Hacker
Renate Sitte

Griffith University, Gold Coast Campus, Parklands Drive, Southport QLD 4215, Australia

This paper describes the development of a computer-based teaching software module called *DigiTrace*, which is designed to simulate the operation of a digital logic circuit by displaying the visual trace of the logic level signals throughout the simulated circuit. The *DigiTrace* module forms an addition to an existing software suite, developed by the authors, for teaching digital logic circuit design. This complete software suite is known as *WinLogiLab* and was initially developed to perform digital logic design. However, feedback from the student users of the software indicated that it would be beneficial to include final simulation capabilities to the software suite. Although the *DigiTrace* module performs similar tasks to existing software for digital logic simulation, the existing software is not compatible with the already designed and successful teaching suite, *WinLogiLab*.

INTRODUCTION

Teaching elementary circuit design can be a challenge as electrical engineering students often do not see the immediate relationship between cause and effect, which can be seen, for example, in mechanical engineering experiments.

Traditionally, the educational system relies on a lecture and whiteboard style of teaching. However, due to the diverse student backgrounds and varying student educational needs, these traditional teaching styles often encounter many educational dilemmas. That is, there is no *best* learning style for all students; each individual student responds differently to varying teaching styles. Educational institutions thus need to provide a broad range of alternative educational resources that the individual student can adapt to their needs.

The changing social attitudes to technology and the increased technical background of students have

resulted in greater computer literacy. Modern students are thus expecting the use of the latest computer-based equipment to prepare them for the modern industrial workplace.

Computer-based learning has therefore become an accepted learning practice that can improve the educational difficulties for both on and off campus students. Furthermore, many educators have reported that the application of multimedia and extensive use of computer-based learning can stimulate a student's interest. The increased student interest can change a student's attitude to a subject, which has proven to greatly improve the student learning process [1-3].

To provide for the subject modernisation and to enhance the learning resources the authors have previously developed a suite of educational software for teaching digital logic design. The software suite is called *WinLogiLab* and it performs the required design techniques for combinatorial and sequential logic circuits as step-by-step tutorials [4-6].

To assess the educational use of the *WinLogiLab* suite, the preliminary modules of the suite were incorporated as part of the digital electronics curriculum in a second year subject. Student feedback indicated the software was educationally useful, beneficial to their studies and that it conveyed the desired information.

*A revised and expanded version of a paper presented at the 4th UICEE Annual Conference on Engineering Education, held in Bangkok, Thailand, from 7 to 10 February 2001. This paper was awarded the UICEE silver award (joint fourth grade with three other papers) by popular vote of Conference participants for the most significant contribution to the field of engineering education.

The feedback also provided valuable suggested improvements [4-6].

A distinct theme in the suggested improvements indicated that it would be beneficial to add capabilities to perform the final testing of the designed circuits. This required a software module that performs circuit simulation. The simulation is performed by displaying the trace of the digital logic signals through the circuit. The tracing of the logic levels through the circuit is sufficient visual indication of the circuit's function.

This paper presents a brief overview of *WinLogiLab* and discusses in more detail the development of the additional software module called *DigiTrace*. The *DigiTrace* module simulates the circuit operation by tracing the logic level signals throughout the user-supplied circuit.

EXISTING DIGITAL SIMULATION PACKAGES

It was initially not considered to be necessary to incorporate logic circuit simulation in the *WinLogiLab* teaching suite. The opinion was that circuit simulation capabilities were redundant in view of the numerous existing and successful commercial and shareware digital logic simulation software available. Some examples of the existing digital simulation software are given below:

- Logic Works [7]
- Logisym [8]
- The SIMIC Logic simulator [9]
- Electronics Workbench [9]
- *WinPSpice* Mixed Mode Simulator [11]

Although these similar commercial and shareware digital logic packages are available, they are not compatible with the Windows based circuit design learning environment *WinLogiLab*. Thus, the circuits produced by the *WinLogiLab* suite do not readily link with these commercial and shareware packages. The developed *DigiTrace* module enables students to directly simulate the designed circuits in *WinLogiLab*, without requiring the additional effort of transferring the designed circuit to alternative simulation software.

OVERVIEW OF WINLOGILAB

WinLogiLab is an MS Windows compatible educational software suite that aids in tertiary, undergraduate digital logic courses. Its application is two-fold because it is firstly suitable for demonstration

purposes during lectures, as well as suitable for individual student self-paced interactive learning, practicing and experimenting.

WinLogiLab Structure

WinLogiLab covers the complete introductory digital logic design process: from combinatorial logic to sequential logic design. That is, the software demonstrates the link between Boolean algebra, truth tables, logic circuits and the minimisation techniques of combinatorial logic. These fundamentals are extended into sequential circuits to design J-K flip-flop counters and Finite State Machines. It is user friendly and thus very intuitive to use and operates while producing formal presentations and realistic design outputs.

WinLogiLab is comprised of a number of software modules or components. The main combinatorial modules are:

- WinBoolean
- BoolTute
- WinEspresso
- DigiTrace

The modules that cover sequential circuits are:

- WinCounter
- StateMac

Figure 1 shows the structure of *WinLogiLab* and the relation amongst its components. Each of these main modules are discussed below.

WinBoolean is designed to show the link between the equivalent digital logic forms of logic gates, Boolean algebra and truth tables. *WinBoolean* allows all three logic forms to be entered, and it can automatically convert the user input to any of the other equivalent forms. An example of this conversion is given in Figure 2.

A graphic facility aids in picking gates from a menu and placing and connecting them to build the circuit diagram. Alternatively, a student may wish to start with an equation (aided by a logic equation editor) and then view the other logic forms.

WinBoolean also provides *logic minimisation*, which enables the input logic to be reduced to the simplest form.

BoolTut is designed to provide an interactive tutorial on logic minimisation techniques. The minimisation can be performed by the Karnaugh map or the Quine-McCluskey process. The program operates with a user-supplied truth table, and displays an animated

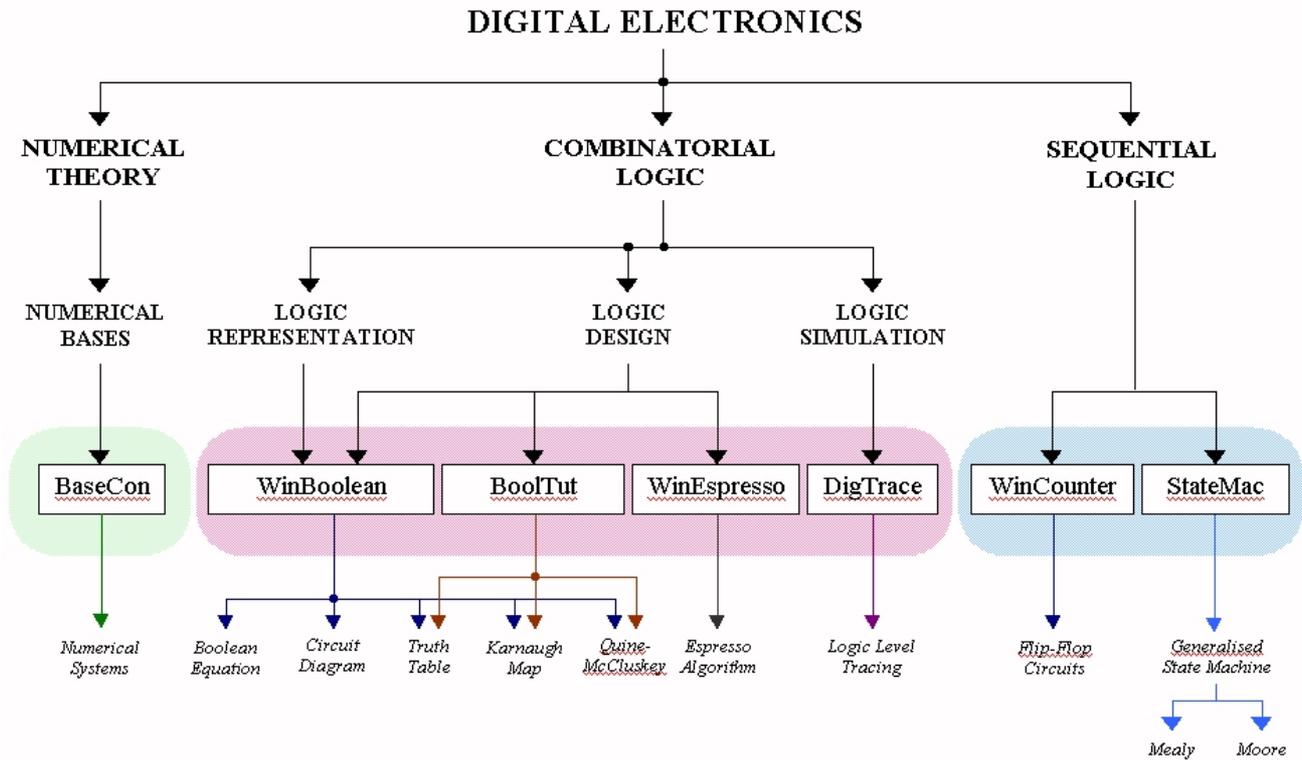


Figure 1: The structure of *WinLogiLab* interactive circuit design teaching/learning suite.

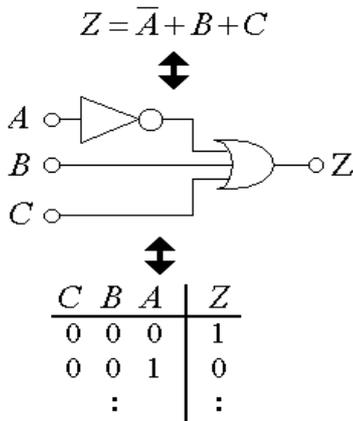


Figure 2: *WinBoolean* allows for interchangeable representations of a logic circuit.

step-by-step presentation on each individual stage of the minimisation process.

WinEspresso provides an alternative method for simplifying Boolean truth tables using a more efficient rule-based approach. The method provides for fast simplification of truth tables with many input and output variables.

WinCounter enables the circuit design, or circuit analysis, of sequential state machine counters. The counters are implemented with J-K flip-flop components and associated combinatorial logic gates.

WinState extends the sequential capabilities of the software suite, by implementing general purpose

Finite State Machines (FSM). *WinState* allows both the design and analysis of Mealy or Moore general purpose FSMs, operating with user-provided input data. A graphic facility aids in designing the states and transitions and their labelling. Once the design is complete, an input string of characters or symbols can be entered to test the FSM. The test is performed by *running* the input string on the FSM. During the run, each transition sequence is stepped through and viewed in animation.

WinLogiLab is MS Windows compatible software that displays in the user friendly graphical user interface (GUI). It has standard MS Windows capabilities and utilises extensive user help.

MS Windows compatibility has the benefit that the user environment is familiar to students. The familiar environment thus minimises the problem of *learning yet another computer program*.

The GUI environment allows the output data to be presented in a form similar to *real life*. Also, the graphical display permits animation and step-by-step display of the theoretical procedures.

The standard MS Windows functions, such as copy and paste, were programmed into the software. This makes it easy for students to copy data from the application, to include in assignments and reports.

A tutorial mode and *help windows* have been implemented in the software. These enable the output of clear instructions that update for each step of the application. This supports the novice user by explaining

what is going on, what inputs are required and what options are available. This steps the user through the application, inviting them to proceed until the final design is completed.

THE DIGITRACE MODULE

The *DigiTrace* software module provides the user input of a digital circuit and displays the visual trace of the logic levels through each component of the circuit. The logic levels are indicated by simulated LED lights. A red light represents a logic low (0), while a green light represents a logic high (1).

There are one to four inputs included within the circuit and these inputs can have varying logic levels. Thus, the logic levels throughout the circuit can be traced through for all possible combinations of the input logic levels. The block diagram of *DigiTrace* is shown in Figure 3.

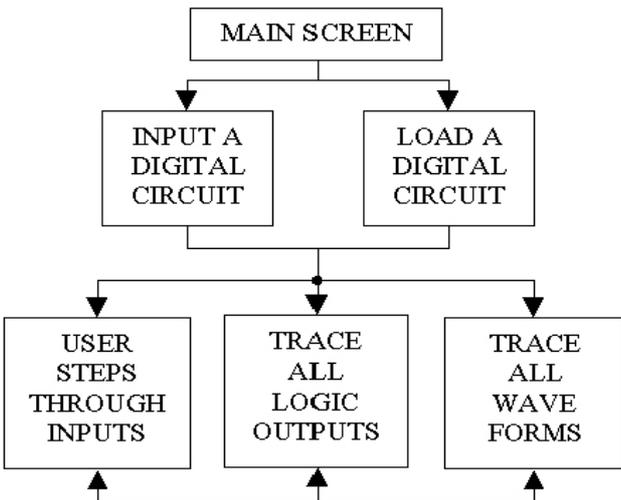


Figure 3: Block diagram of the *DigiTrace* component.

The software can take as input a digital logic circuit designed by one of the other software modules [4-6]. Alternatively, the software allows the user to enter their own circuit schematic, using the toolbar design icons and the mouse pointer. An example screen image of a user input logic circuit is shown in Figure 4. The example also shows the LED lights tracing the logic levels through the circuit.

The user can manually modify the logic inputs by using the mouse to press the buttons available on the input boxes. Alternatively, each consecutive sequence of input logic levels can be stepped through by use of the stepping controls. An example of the controls available to the user to select the current input logic levels is shown in Figure 5.

If the user desires a complete list of all inputs and the corresponding outputs, then the *Trace - ALL*

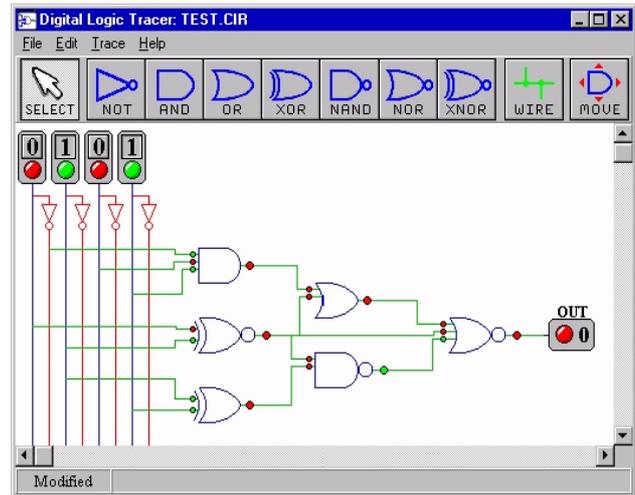


Figure 4: Example circuit input with logic level traces.



Figure 5: Example user selectable logic level input controls.

Inputs menu item will provide this. On this selection the *Trace of ALL Inputs* windows opens to display this list. An example screen image of the window is given in Figure 6.

The Trace ALL Inputs list is actually a different form of the circuit's *truth table*. The truth table is used in the initial design of the logic circuit and thus this list can be used to compare against the original design requirements.

The Trace ALL Inputs provides a list of all the possible inputs with the corresponding output. If the user desires a more complete list of the digital logic levels from each component in the circuit from all the possible input logic levels, then the *Timing* window can be used. An example screen image of the Timing window is given in Figure 7.

Unlike the previous Trace ALL Inputs window, which displays the logic levels as coloured LED lights, the Timing window uses digital square wave pulses. In this case, a peak in the wave represents a logic high (1), and the base line represents a logic low (0).

This pulse trace output is in the conventional square waveform. The square waveform is also the usual output that is obtained by most commercial and shareware logic simulation packages. The Timing window's square wave display was included to allow this modules output to be compatible to the existing simulation software.

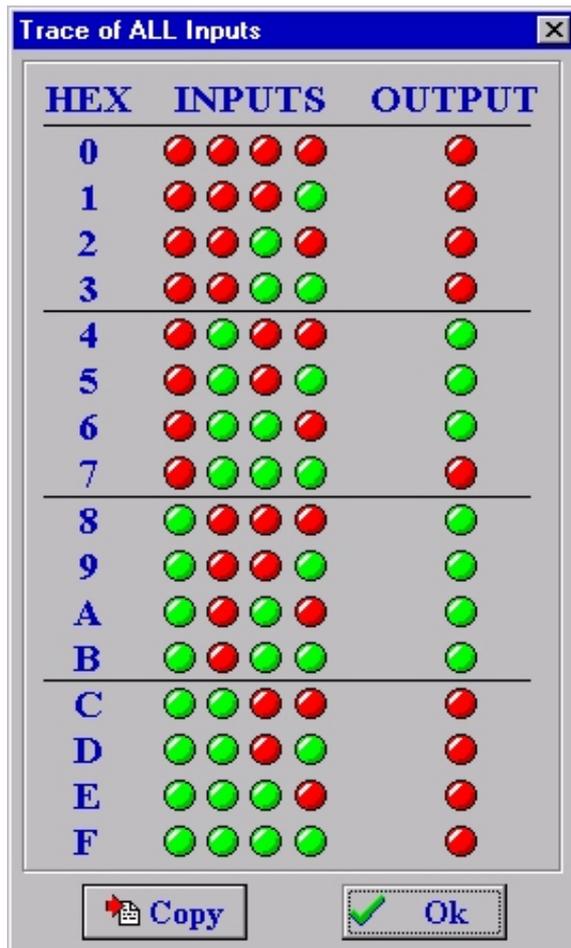


Figure 6: Example of the Trace ALL Inputs window.

CONCLUSION

This paper has presented the development of a software module called *DigiTrace* and forms the final phase of an existing computer-based educational software suite called *WinLogiLab*. The *DigiTrace*

module completes the software suite by providing the final simulation of the digital logic circuit. It has been shown in the paper that the simulation is performed by tracing the logic level signals throughout the user supplied input circuit.

Although this software module performs similar tasks to existing software for digital logic simulation, the existing software are not compatible to the implementation and successfully trialed circuit design learning environment *WinLogiLab*.

REFERENCES

1. Allen, J. and Terman, C., Interactive learning environment for VLSI design. *Proc. IEEE*, 88, 1, January, 96-106 (2000).
2. Huelsman, L., VIEW - Virtual Workbench for Electronic Instrumentation, and life long learning on the World Wide Web. *IEEE Circuits and Devices Magazine*, 16, 5, September, 3-4 (2000).
3. Yang, J. and Anderson, M.. PowerGraf: an educational software package for power systems analysis and design. *IEEE Transactions on Power Systems*, 13, 4, November, 1205-1210 (1998).
4. Hacker, C. and Sitte, R., Development of a computer program to electronically design Digital Logic Circuits using Boolean algebra. *Proc. 9th Annual Australasian Assoc. for Engng. Educ. (AaeE97)*, Ballarat, Australia (1997).
5. Hacker, C. and Sitte, R., A computer based tutorial for demonstrating the solving of digital electronic circuits. *Proc. 10th Annual Australasian Assoc. for Engng. Educ. (AaeE98)*, Gladstone, Australia (1998).
6. Hacker, C. and Sitte, R., Implementing the 'Espresso - Two Level Logic Minimiser' algorithm

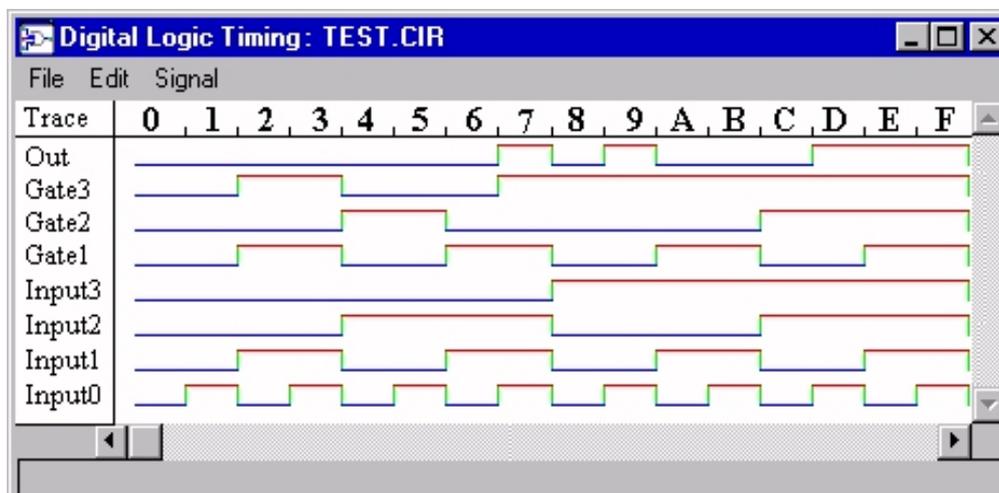


Figure 7: Example of the Timing window.

in the MS-Windows environment. *Proc. 2nd Asia-Pacific Forum on Engng. and Technology Educ.*, Sydney, Australia, 124-127 (1999).

7. Capilano Computing Systems, *Logic Works Version 3.0*. Calif.: Benjamin Cummings Publishing Company (1995).
8. Dobson, A.J., *Logisym Version 3.0* (ShareWare). WinSite Software Repository (1992).
9. Genashor Corporation, *The SIMIC Logic simulator - Version 1.00.04*. (Student Version). <http://tebbit.eng.umd.edu/simulation/comp.simulation/simic.html> (1993).
10. Interactive Image Technologies, *Electronics Workbench 3.0E*. Distributed by Applied Electro Systems Australia, <http://www.interactiv.com> (1993).
11. Orcad/MicroSim Corporation, *WinP Spice Mixed Mode Simulator - Version 9* (Student Version). <http://www.orcad.com> (1999).

BIOGRAPHIES



After obtaining Electrical Engineering and Applied Science qualifications at the University of Central Queensland (UCQ) in 1990), Mr Charles Hacker started working in the UCQ Physics Department as a demonstrator, tutor and sessional lecturer. Mr Hacker later

obtained a position as lecturer in electronics with the School of Engineering at Griffith University, Australia. Mr Hacker went on to complete a Graduate Diploma in Medical Physics at the Queensland University of Technology in 1995. Mr Hacker is currently undertaking an MPhil at Griffith University, researching the area of computer-based teaching of electronics. His teaching areas include electronics, microprocessors, physics, and computer programming.



Renate Sitte obtained her PhD (1995) and MPhil (1993) from Griffith University in microelectronic engineering. She received her Systems Engineering degree (Ingeniero de Sistemas) in 1985 from the Universidad de Los Andes in Venezuela. She is a faculty staff in Engineering and Information

Technology, Griffith University, Australia, where she has been teaching computing, software engineering and discrete mathematics since 1986. Her field of research is modelling and simulation; one major ongoing project is Virtual Reality Prototyping Micro-Electro-Mechanical Systems (MEMS). She is Vice-Chair of the IEEE Queensland Section and Chair of the IEEE Computer Chapter. She is also a member of the Institute of Engineers Australia and contributes to the development of the ISO standards for software engineering.