

DigiSlim: A learning tool for supplementing introductory courses in digital electronics

SIDDHARTH BATRA AND SANJAY GOEL
Jaypee Institute of Information Technology, India

Students studying undergraduate courses in digital electronics often lose interest in the subject, due to the shortcomings of learning tools that are commonly used to supplement such courses. Ideally learning tools increase student participation in the learning process and help in overcoming difficulties inherent to the subject. Comparisons have been drawn between the existing learning tools. DigiSlim, the tool discussed in this paper provides a user friendly environment which empowers novices to create and simulate combinational as well as sequential circuits. Circuit simulations are facilitated through a novel combination of integrated progressive timing diagrams, dynamic value updates of selected circuit points, and dynamic circuit visualization. Existing circuits can also be encapsulated into smaller digital components to enable creation of more complex circuits. It has several other ancillary utilities such as a circuit / IC database and error handling.

Categories and Subject Descriptors: K.3.1 [**Computers and Education**]: Computer Uses in Education

General Terms: Algorithms, Design, Documentation

Additional Key Words and Phrases: Digital circuits, e-learning, interactive simulations, integrated circuits.

1. INTRODUCTION

Since the origin of digital electronics as a course, it has always been supplemented with learning aids. Earlier, these learning aids were in the form of cumbersome laboratory hardware but with the increasing usage of computers to supplement education, the focus of

Author's address: Siddharth Batra and Sanjay Goel, Computer Science Department, Jaypee Institute of Information Technology (JIIT) A-10, Sector-62 Noida, 201307 Uttar Pradesh, India; email: siddharthbatra@gmail.com and sanjay.goel@jiit.ac.in; WWW: <http://www.jiit.ac.in/>

learning aids has shifted towards simulation based learning tools. There have been a handful of simulation tools designed for digital electronics; however few of them possess the right combination of simplicity and user empowerment.

The evolution of the computer into a rich, visual and interactive medium creates several avenues for its utilization within the field of digital electronics. Computer Based Learning (CBL) environment for introductory courses in digital electronics [Da Bormida et al. 1997] is founded on the principles of expository, demonstrative, interactive, and practical pedagogical methodologies. Special emphasis is put on the progressive use of simulation techniques to stimulate learner activity. *EasySim* [Research Systems Pty. Ltd. 2006] is a simulation tool which allows easy creation and modification of circuits, with features such as visual indicators and context sensitive help. Since *EasySim* is a commercial product, it is beyond the reach of many educational institutions and students. A time domain digital simulator titled *6.111 C Simulator* [Eichin 1994] created by the Massachusetts Institute of Technology (MIT) empowered the user by providing a set of macros and libraries in the C programming language. However, the cumbersomeness of this tool makes the simulator overwhelming for beginners. Later, MIT presented another electronics design automation software [Knaian 1998] which is freely available to educational institutions and students. This software titled *Digital Simulator* simulates an array of digital circuit elements, and presents an easy to use “paint” approach to draw circuits. Although the software is free, it lacks in-depth features. Some web based learning environments like *Digital WorkShop* [Lynch and Fishwick 1996] and *Circuit Builder* [Karweit 1997] have limitations of visual quality, bleak user interface, limited user interactivity, modest feature sets, or restricted circuit size.

The simulation tool titled DEEDs [Donzellni and Ponta 2003] presents a learning environment for digital electronics, with tools and resources for teachers and students. It supports combinational and sequential logic networks, finite state machines, microcomputer

interfacing and programming at assembly level. All these modules follow a “learn-by-doing” approach, and, being fully integrated together, they allow design and simulation of complex networks including standard logic, state machines and microcomputers. The tool integrates within an HTML browser and enables internet navigation to find pages with lessons, exercises and laboratory assignments. It also includes an assistant browser to provide step-by-step assistance to students. It has been specifically designed to support distance education and project work within an inter-institutional and international context. In spite of all its positive features *DEEDS* is far too cumbersome to be used for supplementing an introductory course in digital electronics for the simple reason that, its complex user interface and in depth features can overwhelm beginners.

B2 Spice [Beige Bag Software 2006] and *Logisim* [Burch 2002] are the only simulation tools which allow the creation of new parts from an existing circuit. These parts can further be used to create more complex circuits. *B2 Spice* also introduces interactive simulations which allow a lot more user participation during circuit simulations. However, it has a very complex interface and it is also a commercial product which does not make it easily accessible to students and educational institutions. *Logisim* is based on the Java Virtual Machine which enables it to work on a variety of platforms. However, it lacks support for sequential circuits which is a major drawback as sequential circuits form a major part of all introductory courses in digital electronics.

The following parameters are identified for the comparison of various existing Computer Based Learning tools for supplementing introductory courses in digital electronics:

- 1 Ease of use
- 2 Rich GUI
- 3 Circuit simulation
- 4 User interaction during simulations
- 5 Ideal feature set (Not too complex and not too naïve)
- 6 Web based
- 7 Free availability

- 8 Open source code
- 9 Allows creation of Integrated Circuits (ICs)
- 10 Does not require learning a tool specific language.

These parameters are chosen to reflect the requirements of undergraduate students, studying introductory courses in digital electronics. During the study of various existing tools some of the major drawbacks found were, the use of a tool specific programming language for designing circuits [Eichin 1994] and complex in-depth features [Donzellni and Ponta 2003] seldom required by beginners. A lack-luster and cumbersome user interface combined with the lack of user interactivity makes some of these tools [Lynch and Fishwick 1996; Karweit 1997; Eichin 1994; Karweit 1997] very difficult to use for a beginner. Furthermore, the commercial nature of some simulation tools [Research Systems Pty. Ltd. 2006; Beige Bag Software 2006] puts them beyond the reach of students and educational institutions. Table I gives an overview of the comparison of existing tools in the light of the earlier mentioned ten parameters.

Table I. Parameter wise comparison of learning tools for digital electronics

Learning Tool / Parameters	1	2	3	4	5	6	7	8	9	10
6.111 C Simulator			Y				Y	Y		
B2 Spice		Y	Y	Y					Y	Y
Circuit Builder						Y				Y
DEEDS		Y	Y	Y		Y	Y			Y
Digital Simulator			Y				Y	Y		Y
Digital Workshop	Y		Y			Y	Y			Y
EasySim	Y	Y	Y							Y
Logisim		Y	Y				Y	Y	Y	Y

The majority of present day computer based learning tools designed for the field of digital electronics are too cumbersome, hence beginners find them difficult to adapt to. They lack

user friendliness in their interface and simplicity in their operations which turns an otherwise thought provoking user experience into a burden. DigiSlim, the tool discussed in this paper provides a user friendly environment which empowers novices with enhanced learning opportunities.

2. DIGITAL ELECTRONICS LAB SIMULATOR (DIGISLIM)

DigiSlim allows the user to create, compile, and simulate combinational and sequential circuits. The user can save and load circuits or ICs. The user can also create custom ICs and use them in more complex circuits or add them to the predefined database of ICs. Branches on wires, color coordinated outputs, a high level of user interactivity during simulations, a user friendly interface, and progressive timing diagrams for sequential circuits are some of the features of DigiSlim. This section discusses some fundamental functionalities like circuit creation, error handling, the default IC library, and circuit simulation.

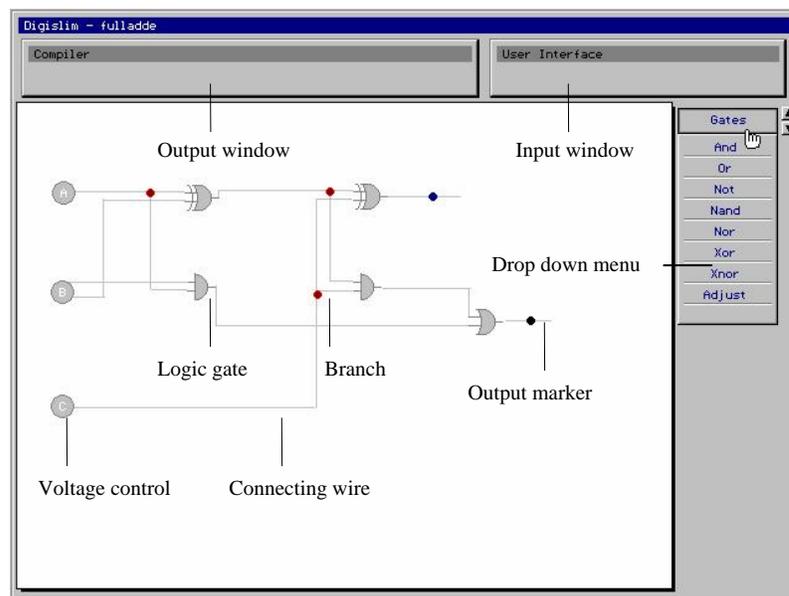


Fig. 1. GUI Description

2.1 Circuit Creation

The majority of the available desktop real estate is segregated for circuit associated functionalities such as creation, modification and interactive simulations (see Figure 1). The circuit creation module offers several functionalities that enhance the circuit designing experience of the user.

2.1.1 Gates / ICs. Logic gates and integrated circuits are the building blocks of all digital circuits. They are both handled in a similar manner during the creation and modification of a circuit. A wide variety of either of them can be selected from a drop down menu. After selection of a digital component, the user can customize everything ranging from the size of the component, to the direction in which it must face. The number of inputs for basic logic gates can also be configured. A “drag and drop” functionality is used to allow the user to place the component anywhere on the circuit creation area. Figure 2 illustrates positional change of a component without breaking any existing connections to other components. These digital components can also be removed from the circuit using a “click to remove” functionality.

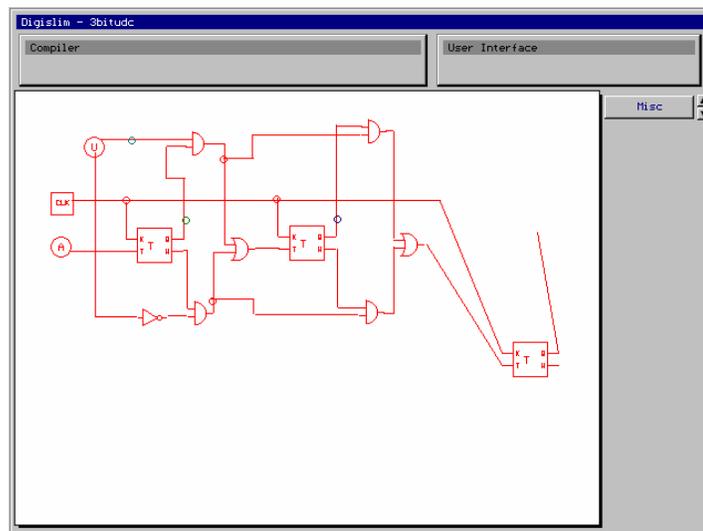


Fig. 2. Modifying the location of an IC

2.1.2 *Wires.* The digital components within a circuit must be connected. Connecting wires are created using a “drag and draw” functionality. There is also an added functionality for drawing horizontal and vertical wires. This helps in reducing the clutter in a circuit. While drawing wires, it is not necessary to link wire ends down to the exact pixel. The circuit compiler searches for adjoining links within a specified radius from a wire end. This eases the usability of the component joining phase in the circuit creation process, as components linked within the bounds of error are also considered connected. Figure 3 shows the process of modifying a wire by splitting it along its length and then relocating it.

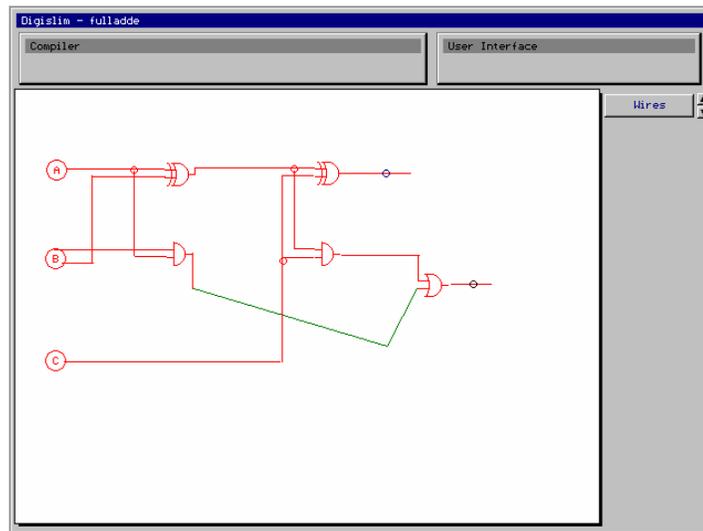


Fig. 3. Modifying the location and structure of a wire

2.1.3 *Inputs Voltage Controls.* An Input Voltage Control is the only mechanism using which the user can control the operation of a circuit. To maintain the simplicity of DigiSlim, voltage controls can only switch between a one and a zero level. Figure 1 shows the usage of voltage controls in creating a full adder circuit.

2.1.4 *Branches.* A large number of wires are utilized during the creation of complex circuits. In such a scenario, it is cumbersome to connect each wire directly to a voltage control. In order to maintain the visual clarity of a circuit, a facility to connect branches

anywhere along the length of an existing wire is provided. A branch holds the same value as that of the wire from which it originates. Branches help in reducing the number and length of wires required to create a digital circuit. Branches also help in reducing the number of gates and ICs required to create a circuit. The circuit shown in Fig 4 has been created without the use of branches. Although the circuit is operationally correct, the general disorganization and frequent overlapping of wires make it difficult to understand. In contrast, the circuit shown in Figure 5 has been drawn using branches.

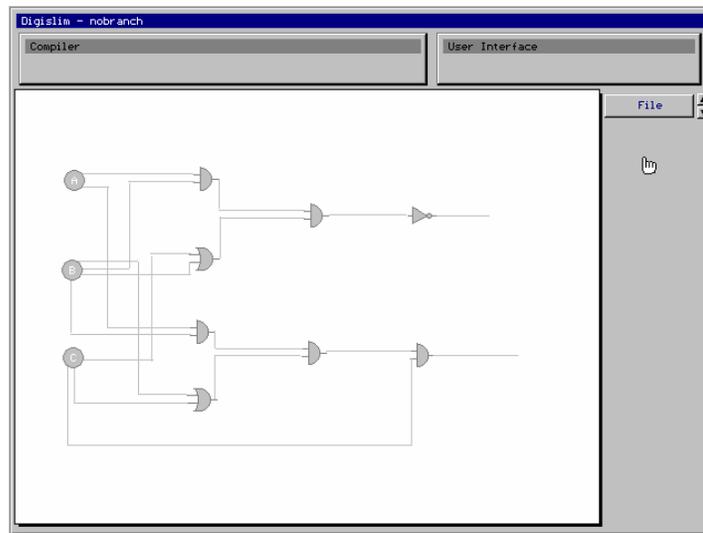


Fig. 4. A digital circuit created without the use of branches.

The circuit in Figure 5 is much neater and hence easier to understand. The use of branches has also reduced the number of logic gates and wires required to perform the same operation. Branches are even more useful in complex circuits where organization, neatness and conservation of desktop real estate are crucial. Both the circuits shown in Figure 4 and 5 perform the same operation but the difference in their organization is evident.

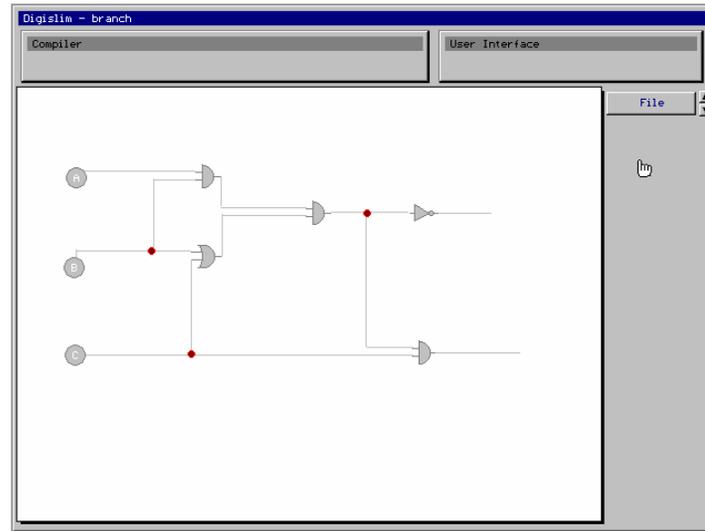


Fig. 5. Modification of the circuit shown in Fig.4 by the use of branches

2.2 Error Handling

During the creation of circuits, the user may make errors while joining digital components.

The compilation phase of DigiSlim detects and reports three types of logical errors: feedback errors, broken connections, and multiple input allocation errors.

2.2.1 Feedback errors. During the creation of a combinational digital circuit, the user may create a deadlock situation. A deadlock situation or a feedback error occurs when inputs of two or more logic gates are dependant upon each other's outputs. Figure 6 shows a combinational circuit with a feedback error. This error, if not handled shall result in an infinite waiting period during the evaluation phase for combinational circuits.

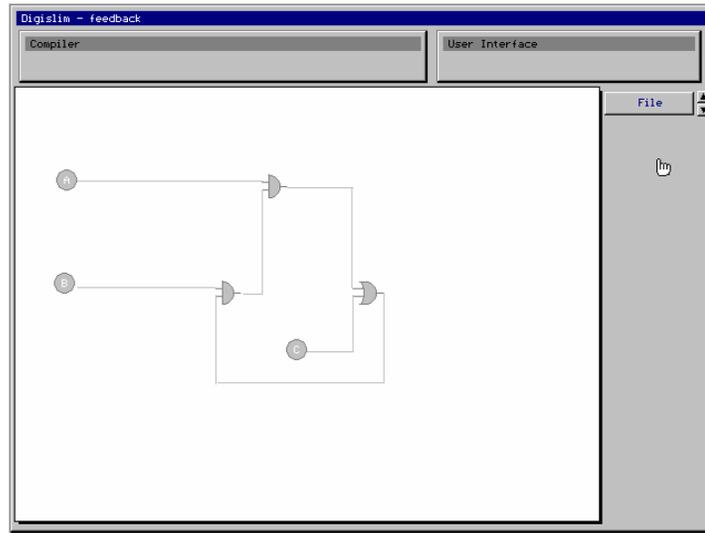


Fig. 6. A deadlock situation (Feedback error) in a combinational circuit

2.2.2 *Broken connections.* Connections in which, a wire end has not been properly placed in the vicinity of a digital component are called broken connections. They are equivalent to a physical break in the circuit. Figure 7 shows a broken connection detected by circuit compiler.

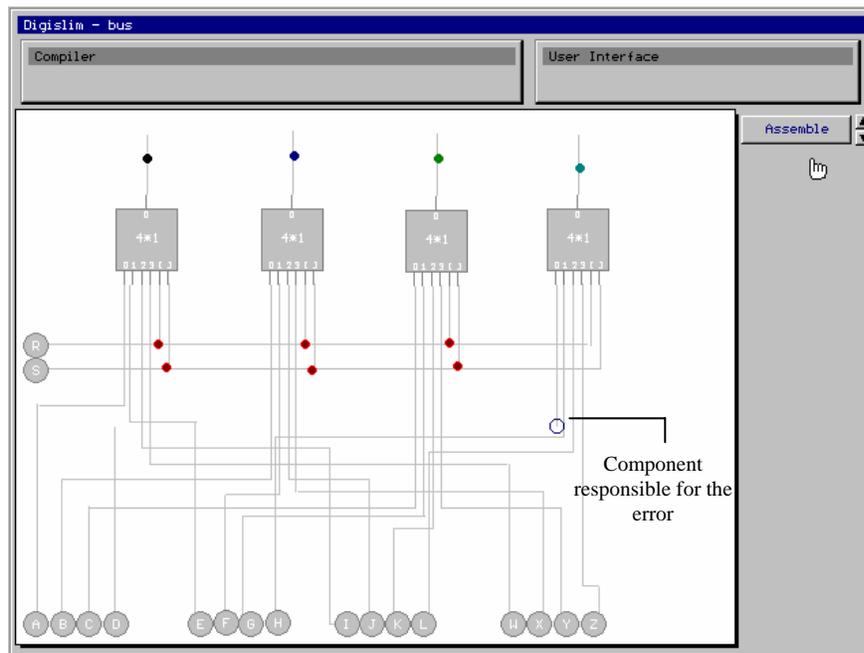


Fig.7. Error handling for broken links during the compilation phase

2.2.3 Multiple Input Allocation Errors. It is incorrect to connect more than one wire to an input of a logic gate or an IC. An error resulting from this condition is known as a multiple input allocation error. This type of error commonly occurs during the creation of large complex circuits where, wire ends may overlap due to human error.

The circuit compiler detects all afore mentioned errors and directs the user's attention towards these detected errors by suitably highlighting the relevant component(s). The user then proceeds to debug the error reported. The user can simulate and create integrated circuits after successful compilation.

2.3 Default IC Library

Integrated Circuits encapsulate the functionality of an entire circuit and aid in the creation of uncluttered complex circuits. DigiSlim provides a predefined database of commonly used Integrated Circuits. These ICs can simply be dragged and dropped onto the circuit creation area and can be utilized in the creation of complex circuits. There is also an option for encapsulating several ICs within an IC. As the creation and storage of these ICs is based on truth tables, the speed at which the output is evaluated is completely independent of the complexity of encapsulated ICs.

2.4 Circuit Simulation

Simulation techniques stimulate learner activity, which lead to an increase in user participation in the learning process. DigiSlim provides specific simulation techniques to support combinational as well as sequential circuits. During the simulation of these circuits, the user can toggle the input voltage controls (between a zero and one level) and observe the corresponding changes made to the circuit. Figure 8 shows a simulation of a combinational circuit.

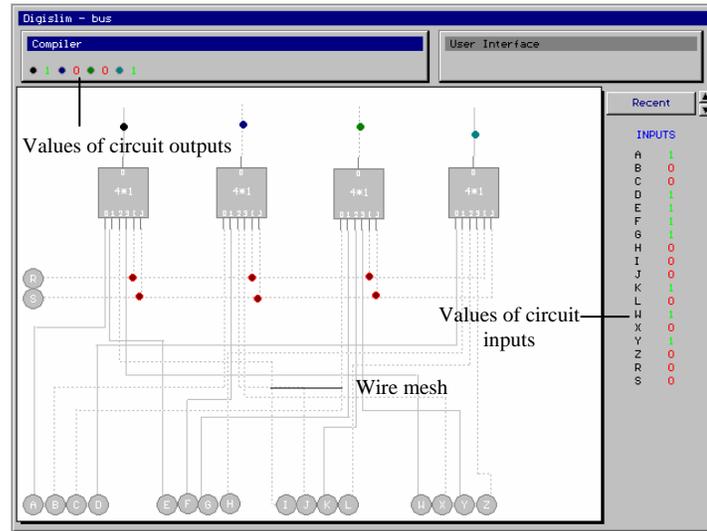


Fig. 8. Circuit simulation for a combinational bus circuit

During the simulation, all the current values of the voltage controls in the circuit are regularly updated. The simulator dynamically updates the style of wires where, dotted wires are used for depicting '0' and solid wires are used for depicting '1'. This dynamic circuit visualization technique helps in analyzing the workings of a circuit at a single glance. The simulator also dynamically updates the values of selected circuit points. These features are especially useful in the case of sequential circuits.

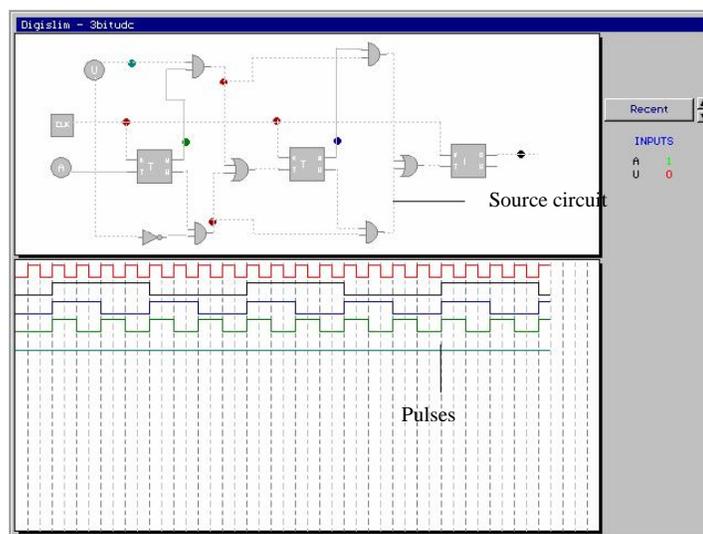


Fig. 9. Progressive timing diagram for a sequential binary up down counter circuit

Sequential circuits are better understood via the use of timing diagrams that display the operation of a circuit in terms of clock pulses. DigiSlim facilitates the display of progressive timing diagrams. Figure 9 shows a simulation stage for a binary up down counter. Each output marker in the circuit is represented by a pulse of the corresponding color. These pulses can be manipulated by toggling the input voltage controls of the circuit. Along with the progressive timing diagram, the learner can simultaneously view the circuit and its progressive timing diagram. This aids in the analysis of sequential circuits.

3. OUTLINE OF ALGORITHMS

This section gives a brief description of the main algorithms which were used to create DigiSlim. The main algorithms which have been discussed are circuit compilation and creation of integrated circuits.

3.1 Circuit Compilation

After the user creates a digital circuit or chooses to load one from the database, the circuit must be compiled before a simulation can begin. The compilation ensures that all digital components have been properly connected without any errors. In case of any error, the compiler alerts the user and highlights the relevant component. The compilation of a circuit works on a very simple technique of iteratively identifying ready components within a circuit. A ready component signifies a digital component whose inputs are all either directly connected to an input voltage control or originate from digital components whose values have been calculated in previous iterations. Hence, a given digital circuit is compiled by iteratively identifying the ready components in the circuit and then calculating their output values by using suitable truth tables.

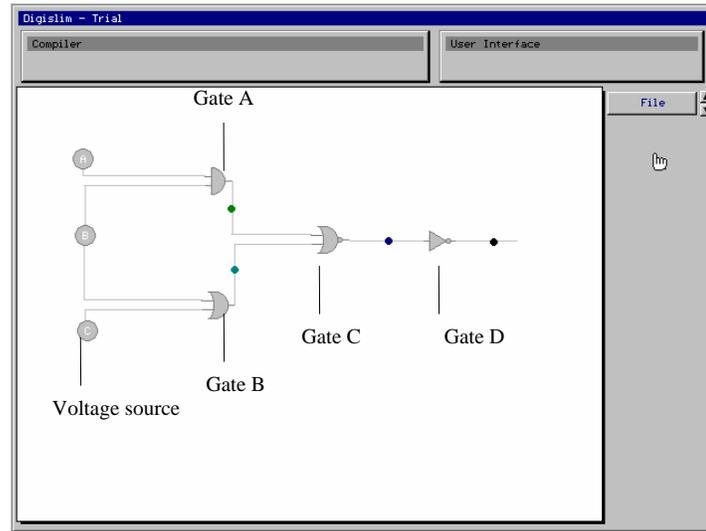


Fig. 10. A sample circuit for illustrating circuit compilation

Figure 10 shows a circuit which shall take three iterations to be compiled. In the first iteration, Gates A & B are the ready components and are hence evaluated using truth tables. In the second iteration, Gate C is the ready component. Gate D is the ready component in the final iteration. Thus using this technique, the most complex of circuits can be evaluated and corresponding errors identified.

3.2 Creation of Integrated Circuits

A circuit may contain several ICs and computational effort for individual components will increase the overall time requirement drastically. Integrated Circuits encapsulate the functionality of an entire circuit into a smaller component and aid in the creation of uncluttered complex circuits. Evaluation of an Integrated Circuit takes lesser time than evaluating the circuit whose functionality it encapsulates. A database of Integrated Circuits is created and maintained to simplify this process. Upon the creation of an IC, its entire truth table is calculated and entered into the database. During compilation, the compiler converts the values of the binary inputs into an integer and retrieves the corresponding outputs from the database table. This ensures a fast computation rate and is only limited by the time

needed to access the database. Figure 11 shows a full adder circuit with its corresponding IC.

This small sized IC allows the user to create complex uncluttered circuits.

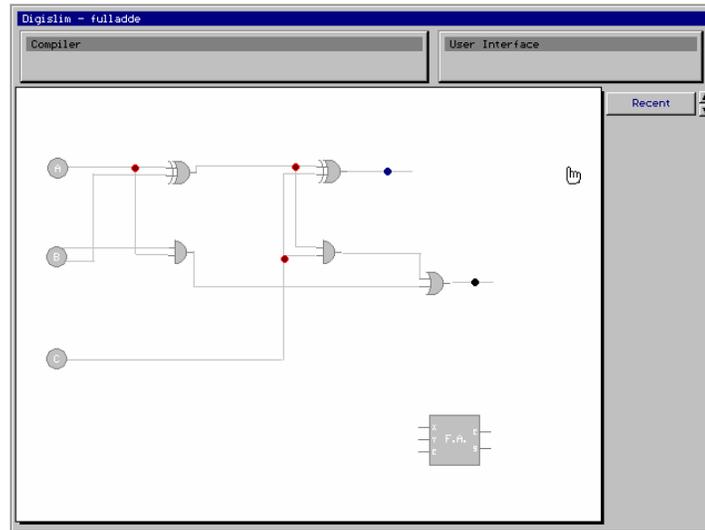


Fig. 11. A full adder circuit with its corresponding IC

4. CONCLUSION

We have evaluated some of the existing CBL tools for Digital Electronics on parameters of ease of use, rich GUI, circuit simulation, user interaction during simulations, ideal feature set, web based, free availability, open source code, creation of integrated circuits, and freedom from learning a tool specific language. We have reported a simulation based learning tool for Digital Electronics. The tool's main utilization is in the form of a learning aid for students studying introductory courses in digital electronics. DigiSlim provides a set of rich features which overcome these shortcomings of existing simulators. Circuit simulations are facilitated through a novel combination of integrated progressive timing diagrams, dynamic value updates of selected circuit points, and dynamic circuit visualization. Existing circuits can also be encapsulated into smaller digital components to enable creation of more complex circuits. It has several other ancillary utilities such as a circuit / IC database and error handling. Further, DigiSlim is free and open source software. Except Web-support, the

current version of Digislim evaluates positively on all other nine out of ten evaluation parameters proposed by us.

5. FUTURE SCOPE

The automated circuit generator module of DigiSlim is still in its developmental phase. This module aims to create a digital circuit from a given Boolean expression. This requires placement algorithms to facilitate the placement of digital components such that they utilize minimum desktop real estate. It also requires routing algorithms to ensure that the wires drawn to connect these ideally placed components utilize minimum wire length and result in minimum overlapping amongst themselves.

REFERENCES

BEIGE BAG SOFTWARE. 2006 (Last Modified). B2 Spice. See

http://www.beigebag.com/adv4_pro.htm. Last accessed 18 May 2006.

BURCH, C. 2002. Logisim: A graphical system for logic circuit design and simulation.

Journal of Educational Resources in Computing 2:1, pages 5-1s6 (2002). See

<http://ozark.hendrix.edu/~burch/logisim/>. Last accessed 4 May 2006.

DA BORMIDA, G., PONTA, D., AND DONZELLNI, G. 1997. Methodologies and

Tools for Learning Digital Electronics. *IEEE Transaction on Education* 40

(November, 1997). See <http://www.sosask.ieee.ca/soc/es/Nov1997/08/INDEX.HTM>

DONZELLNI, G., AND PONTA, D. 2003. DEEDS: E-Learning Environment for Digital

Design. *European Network on Intelligent TEchnologies for Smart Adaptive Systems*

(*EUNITE*), *Training and Education* (July 2003).

EICHIN, M. 1994 (Last Modified). 6.111 C Simulator. See

<http://www.mit.edu/people/eichin/thesis/usrdoc.html>. Last accessed 23 April 2006.

KARWEIT, A. 1997 (Last Modified). Circuit Builder. See

<http://www.jhu.edu/~virtlab/logic/logic.htm>. Last accessed 10 May 2006.

KNAIAN, A. Digital Simulator. 1998 (Last Modified). See

<http://web.mit.edu/ara/www/ds.html>. Last accessed 31 May 2006.

LYNCH, D., AND FISHWICK, P. 1996 (Last Modified). Digital Circuit Simulation on

the Web (Digital WorkShop). See <http://www.cise.ufl.edu/~fishwick/dig/dlesp.htm>.

Last accessed 25 April 2006.

RESEARCH SYSTEMS PTY. LTD. 2006 (Last Modified). EasySim Digital Logic

Simulator. See <http://www.research-systems.com/easysim/easysim.htm>. Last accessed

20 April 2006.