

DIGISLIM: A LEARNING TOOL FOR LOGIC LEVEL DIGITAL ELECTRONICS

Siddharth Batra and Sanjay Goel

Computer Science Department

Jaypee Institute of Information Technology University (JIIT University)

Abstract

DigiSlim, our logic level digital electronics learning tool, provides a user friendly environment which empowers first year undergraduate engineering students to design and simulate combinational as well as sequential circuits. Logic simulations are facilitated through integrated progressive timing diagrams, dynamic value updates of selected circuit points, and dynamic circuit visualization. Existing logic level circuits can be encapsulated into compact digital components to design more intricate logic. It provides ancillary utilities such as a digital component database and error handling. DigiSlim has been assessed by introducing it as a learning aid for the targeted group of students.

Introduction

Introductory courses in digital electronics focus on the logic level rather than circuit, device or layout level representation. 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 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 have the right combination of simplicity and user empowerment. Many tools overwhelm the user with transistor level circuit representation. DigiSlim, the tool discussed in this paper, provides a user friendly environment which empowers novices with enhanced learning opportunities by focusing on logic level representation without overwhelming the user. It has been developed at the Computer Science &

IT department at the Jaypee Institute of Information Technology University and caters specifically to introductory undergraduate courses in digital electronics. DigiSlim is an open source Windows based tool and is freely available at

<http://www.jiit.ac.in/jiit/files/projects/csit/ComputersinSociety/computerscienceeducation/digislim.htm>.

Through its simple and interactive approach towards logic level circuit visualization DigiSlim, along with being a tool for independent-study, can also be used to supplement lectures and laboratories. This Computer Based Learning (CBL) environment for introductory courses in digital electronics [5] 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 [13] is a simulation tool which allows easy creation and modification of circuits, with features such as visual indicators and context sensitive help. Unfortunately, *EasySim* is a commercial product, whose cost is beyond the reach of many educational institutions and students. A time domain digital simulator titled *6.111 C Simulator* [7] empowers the user by providing a set of macros and libraries in the C programming language. However, the tool is cumbersome and makes the simulator overwhelming for beginners. A similar electronics design automation software,[9] which was freely available to educational institutions and students, was later presented. 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* [10] and *Circuit Builder* [11] have limitations of visual quality, bleak user interface, limited user interactivity, modest feature sets, or restricted circuit size.

The simulation tool titled DEEDs [3] 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 [6] and *Logisim* [4] 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.

We have surveyed a combination of commercial and freely available Computer Based Learning circuit simulators in light of the design theory of Minimalism [1]. The Minimalist theory is a framework for the design of instruction, especially training materials for computer users. A minimalist design is expected to leverage and focus only on the most fundamental and essential features. Apart from being extensively used as a guide for instructional design [2], the Minimalist theory has found several applications in the design of web applications [8] and software documentation [2]. *DigiSlim* has been designed via the minimalist approach and attempts to provide the ‘essential feature set’ to the targeted group of students. The tools surveyed are commonly used by academic institutions for supplementing courses in digital electronics and the following parameters have been identified for their comparison:

1. Visual synthesis of circuits
2. Interactive Logic simulation
3. Allows creation of Integrated Circuits (ICs)
4. Rich GUI
5. Free availability
6. Open source code
7. Web based

These parameters are chosen to specifically reflect the requirements of undergraduate students studying introductory courses in digital electronics. A few features commonly provided by commercial circuit simulators such as circuit level simulation along with propagation delays, are not necessary from the perspective of an introductory course and have not been considered.

The value of open source in educational tools cannot be exaggerated. Such tools allow academic institutions to rewrite small sections of the code to customize it according to their laboratory’s needs and aid in providing better learning opportunities. Thus, ‘Open source code’ was found to be an important parameter in the evaluation of learning tools.

During the study of various existing tools some of the major drawbacks found were: the lack of visual synthesis for logic level circuits [7] and overdependence on transistor level circuit design support [3] seldom required by beginners. A lack-luster and cumbersome user interface combined with the lack of user interactivity makes some of these tools [7, 10, and 11] very difficult to use for a beginner. Furthermore, the commercial nature of some simulation tools [6 and 13] puts them beyond the reach of students and many educational institutions in developing countries. Table I gives an overview of the comparison of existing tools in the light of the earlier mentioned eight parameters.

Digital Electronics Lab Simulator (DigiSlim)

DigiSlim allows the user to create, compile, and simulate combinational and sequential logic circuits using available gates and available 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 features like logic level circuit creation, error handling, the default IC library, and circuit simulation.

Table I. Parameter wise comparison of learning tools with respect to the requirements in introductory digital electronics.

Parameters / Learning Tool	6.111 C Simulator	B2 Spice	Circuit Builder	DEEDS	Digital Simulator	Digital Workshop	Multisim	P-Spice	EasySim	Logisim	LogicWorks	Multimedia Logic	DigiSlim
1. Visual synthesis of logic circuits			Y	Y		Y			Y	Y	Y		Y
2. Interactive Logic Simulation		Y		Y	Y	Y	Y			Y	Y	Y	Y
3. Rich GUI		Y		Y			Y	Y	Y	Y	Y	Y	Y
4. Creation of ICs		Y					Y	Y		Y			Y
5. Free availability	Y			Y	Y	Y		Y		Y		Y	Y
6. Open source code	Y				Y					Y		Y	Y
7. Web based			Y	Y		Y							

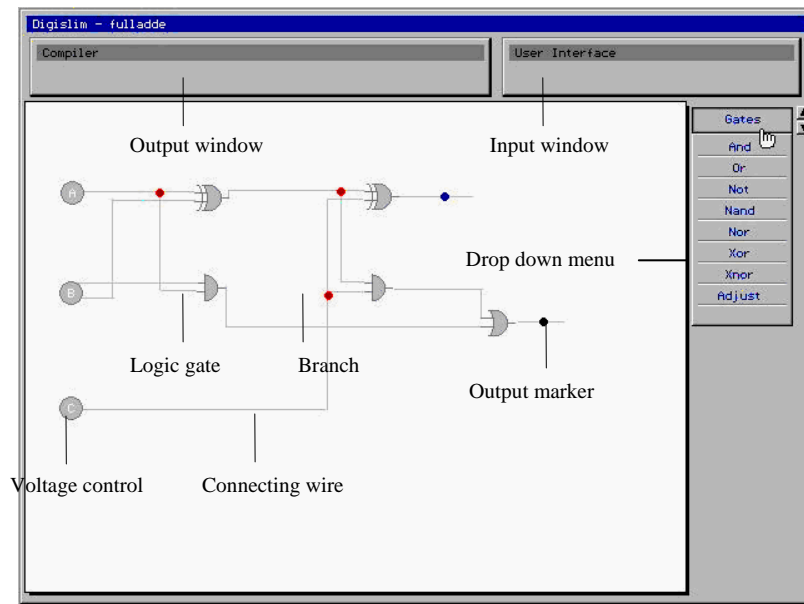


Figure 1. GUI Description.

Logic level 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.

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” metaphor is used to allow the user to place the components. 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 single click.

Wires. Connecting wires are created using a “drag and draw” metaphor. There is an added functionality for drawing horizontal and vertical wires. 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.

Inputs Voltage Controls. This interactive mechanism allows the user to analyze the circuit operation. 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 a full adder circuit.

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

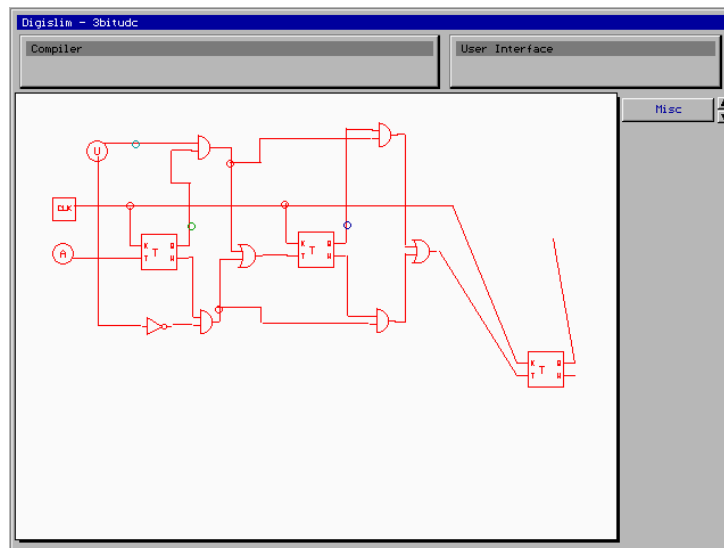


Figure 2. Modifying the location of an IC.

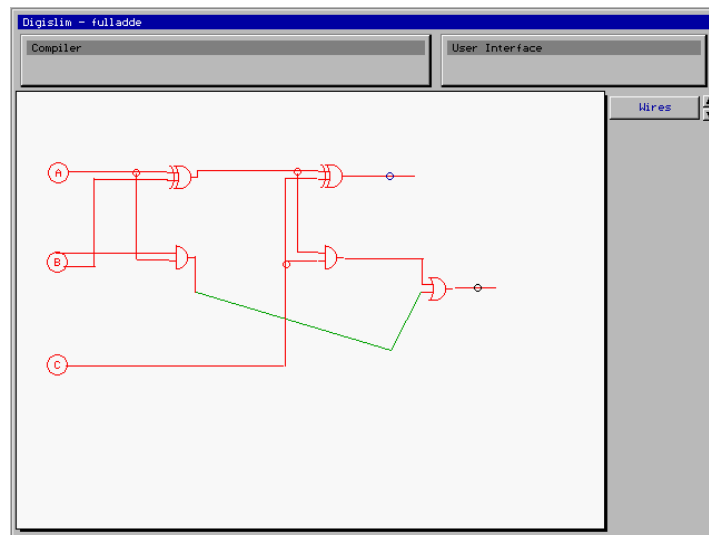


Figure 3. Modifying the location and structure of a wire.

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 for a circuit. The circuit shown in Figure 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.

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 particularly 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.

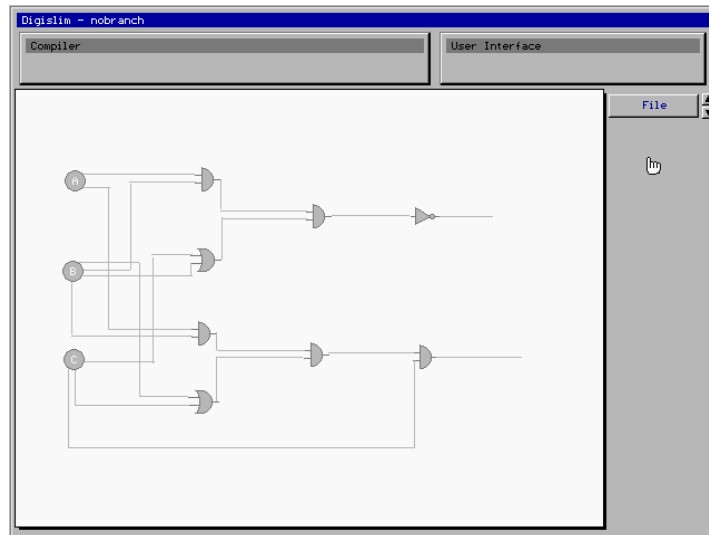


Figure 4. A digital circuit created without the use of branches.

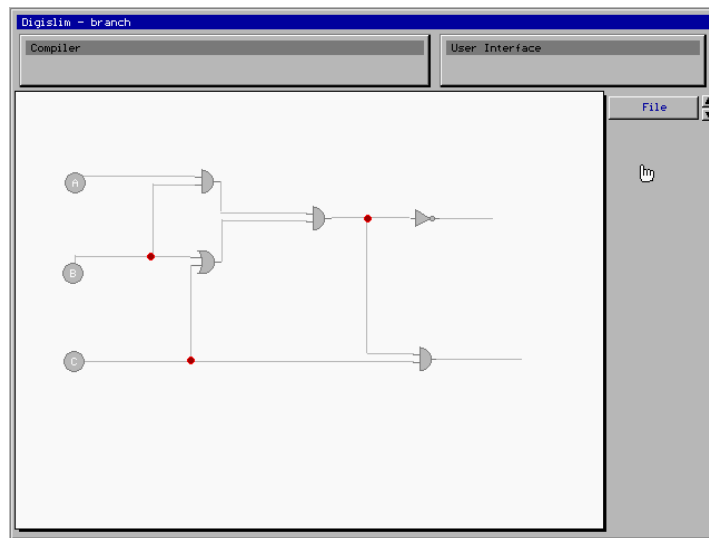


Figure 5. Modification of the circuit shown in Fig.4 by the use of branches.

Error Handling

During the creation of circuits, the user may make errors while joining digital components. The compilation phase of DigiSlim detects and reports two types of logical errors: broken connections and multiple input allocation errors.

Broken connections. If a wire-end has not been properly placed in the vicinity of a digital component, it results into broken connections.

Figure 6 shows a broken connection detected by circuit compiler.

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.

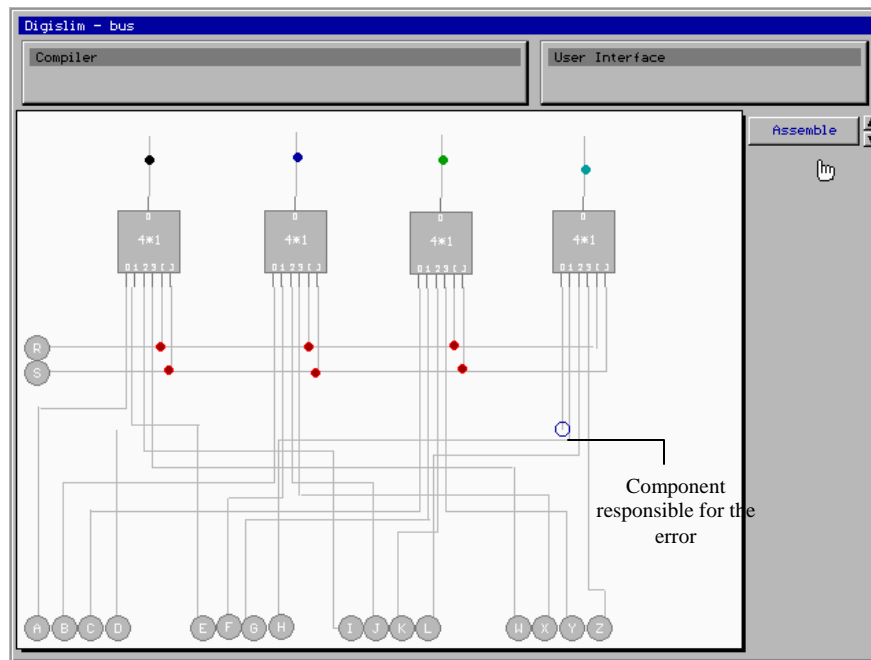


Figure 6. Error handling for broken links during the compilation phase.

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

Default IC Library

Integrated Circuits encapsulate the functionality of a 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.

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 change the propagation delay of components. Figure 7 shows a simulation of a combinational 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. The simulator regularly updates the values of selected circuit points. These features are especially useful in the case of sequential circuits.

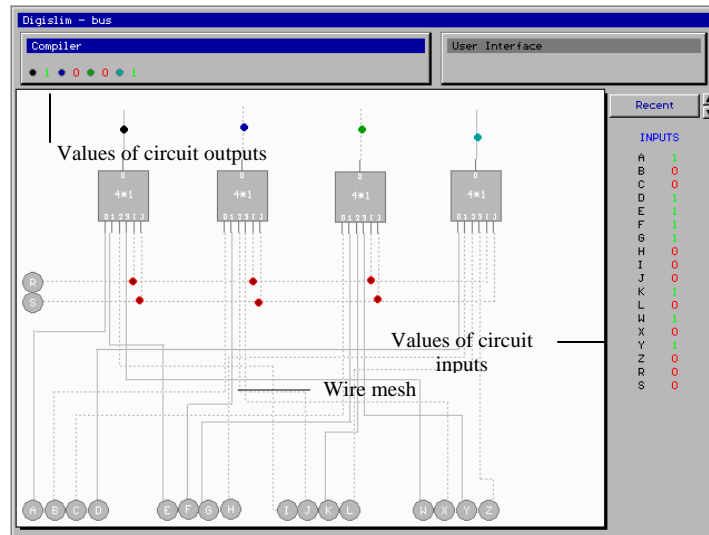


Figure 7. Circuit simulation for a combinational bus circuit.

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 8 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. The learner can simultaneously view the circuit and its progressive timing diagram.

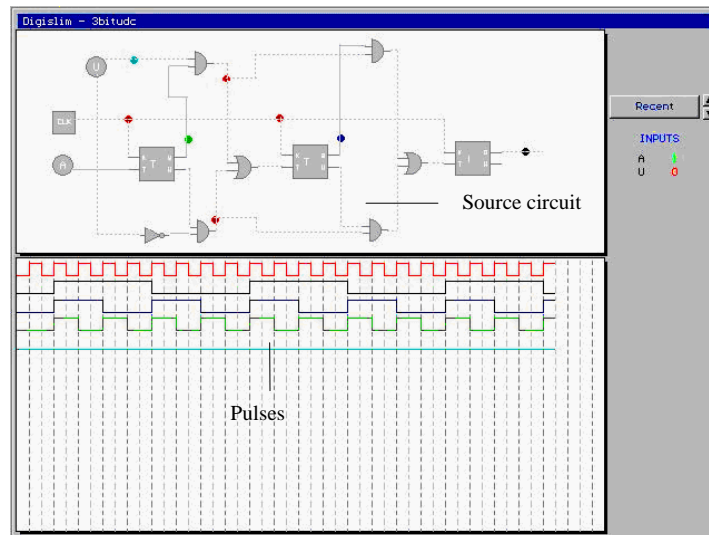


Figure 8. Progressive timing diagram for a sequential binary up down counter circuit.

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.

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 progressively identifying ready components within a circuit and then calculating their output values by using suitable truth tables. 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.

Figure 9 shows a circuit which shall take three iterations for compilation. 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.

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 a 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 truth table is computed 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 10 shows a full adder circuit with its corresponding IC. This small sized IC allows the user to create complex uncluttered circuits.

Results

DigiSlim has been assessed by introducing it as a learning aid for the first year undergraduate course ‘Digital Electronics & Logic Specification’ at the Jaypee Institute of Information Technology University. The students who underwent the experiment were surveyed, to judge the effectiveness of DigiSlim as a supplement for introductory courses. A majority of the surveyed students reported that DigiSlim helped them visualize the concepts and generated an added interest in the subject.

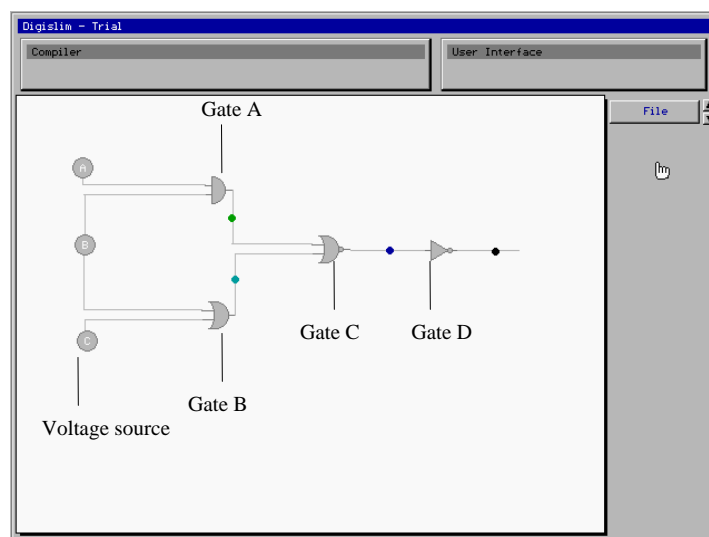


Figure 9. A sample circuit for illustrating circuit compilation.

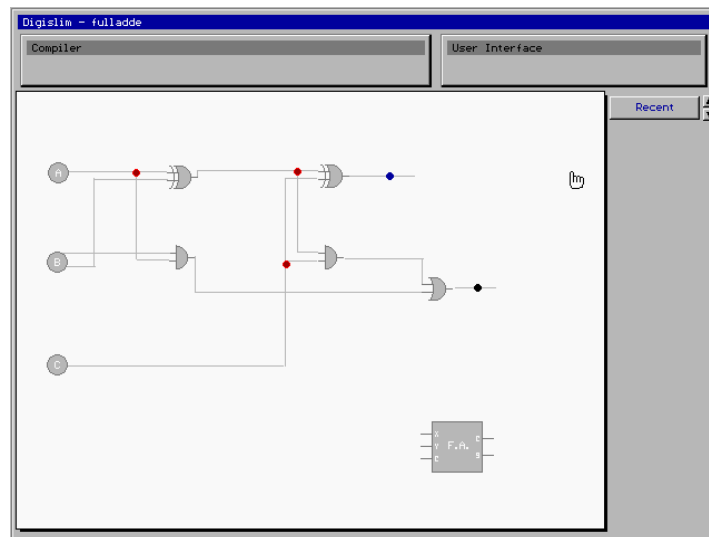


Figure 10. A full adder circuit with its corresponding IC.

Students were also comfortable in working with DigiSlim and found it to be user-friendly especially when compared to the traditionally used circuit simulators which do not provide visualization based methods for simulations. Furthermore, the specifically designed interactive logic simulations for combinational and sequential circuits were reported to be the most useful feature of DigiSlim. Although a small segment of the students criticized the minimalist feature set, the majority of surveyed students welcomed DigiSlim's simple approach.

Conclusion

We have reported a logic level simulation based learning tool for introductory Digital Electronics. We have followed the Minimalist theory to design this tool to overcome the shortcomings of existing simulators. Logic level simulations are facilitated through a 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 digital component database and error handling. Further, DigiSlim is free and open source software. The current

version of DigiSlim was assessed by introducing it as a learning aid for the first year undergraduate course 'Digital Electronics & Logic Specification' at the Jaypee Institute of Information Technology University. The surveyed students liked DigiSlim's approach towards visualizing the concepts of digital electronics, thereby successfully generating an added interest in the subject.

Future Scope

The automated circuit generator module of DigiSlim is still in its developmental phase. This module will 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 wire 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. Multiple-bit input voltage controls and wire combiners and splitters shall also be a valuable addition for managing circuits that require manipulating multiple bit flows. We also plan to empower the user to seamlessly switch between the various representation levels of logic, circuit, device, and layout.

Acknowledgement

We would like to thank Prof. A.B. Bhattacharyya for his feedback.

References

Books

1. J. Carroll, *The Nurnberg Funnel*, Cambridge, MIT Press, 1990.
2. J. Carroll, *Minimalism beyond the Nurnberg Funnel*, Cambridge, MIT Press, 1998.

Conference Papers

3. G. Donzellni, and D. Ponta, "DEEDS: E-Learning Environment for Digital Design", *EUropean Network on Intelligent TEchnologies for Smart Adaptive Systems (EUNITE), Training and Education*, 2003

Journal Papers

4. C. Burch, "Logisim: A graphical system for logic circuit design and simulation". *Journal of Educational Resources in Computing*, 2:1, pages 5-16, 2002
5. G. Da Bormida, D. Ponta, and G. Donzellni, "Methodologies and Tools for Learning Digital Electronics", *IEEE Transaction on Education*, 40, 1997

Websites

6. Beige Bag Software, "B2 Spice", http://www.beigebag.com/adv4_pro.htm, 2006 (accessed 18 May 2006).
7. M. Eichin, "6.111 C Simulator", <http://www.mit.edu/people/eichin/thesis/usrdoc.html>, 1996 (accessed 23 April 2006).
8. I. Pedersen, "Minimalism and HTML-based Help. <http://arts.uwaterloo.ca/~ipederse/Minimalism.htm>, 1997 (accessed 11 Jan 2007).
9. A. Knaian, "Digital Simulator", <http://web.mit.edu/ara/www/ds.html>, 1998 (accessed 31 May 2006).

10. D. Lynch, and P. Fishwick, "Digital Circuit Simulation on the Web (Digital WorkShop)", <http://www.cise.ufl.edu/~fishwick/dig/dlesp.htm>, 1996 (accessed 25 April 2006).
11. A. Karweit, "Circuit Builder", <http://www.jhu.edu/~virtlab/logic/logic.htm>, 1997 (accessed 10 May 2006).
12. Prentice Hall Publishing, "Logic Works 5 Digital Simulation", <http://www.logicworks5.com/index.html>, 2007 (accessed 10 March 2007).
13. Research System Pty. Ltd., "EasySim Digital Logic Simulator", <http://www.research-systems.com/easysim/easysim.htm>, 2006 (accessed 20 April 2006).

Biographical Information

Siddharth Batra is a Master's student in the Artificial Intelligence lab at the computer science department of Stanford University. He is also the CEO of vision based online video startup Zunavision. Siddharth did his BS in computer science from IIIT, India where his research focused on computer vision, graphics and HCI. At Stanford University his research focuses on artificial intelligence and computer vision with a special emphasis on video.

Sanjay Goel is Associate Professor and Head of the Department of Computer Science & Engineering/Information Technology at Jaypee Institute of Information Technology (JIIT), Noida, India. He earned his M.Tech. (Computer Technology) from IIT Delhi, and his B.E (Hons.) (Electrical and Electronics Engg.) and M.Sc. (Hons.) (Physics) from BITS, Pilani. He has nearly 23 years of mixed experience in teaching, software development, and interactive multimedia content creation. Prior to joining JIIT, he was Director (Multimedia) at Indira Gandhi National Centre for the Arts (IGNCA). He has been a faculty member at Delhi University for more than a decade, and has also worked at National Informatics Centre, New Delhi. He enjoys inter-disciplinary work and his professional interests include Learning Sciences & Technology, Software Design, Visual Computing, Multimedia, and Cultural Heritage Informatics.