LoGen – Generation and Simulation of Digital Logic on the Gate-Level via Internet

Stephan Kubisch, Ruediger Rennert, Hartmut Pfueller, Dirk Timmermann

University of Rostock
Department of Computer Science and Electrical Engineering
Institute of Applied Microelectronics and Computer Engineering
18051, Rostock
Germany
{stephan.kubisch,ruediger.rennert,hartmut.pfueller,dirk.timmermann}@uni-rostock.de
http://www-md.e-technik.uni-rostock.de

***Abstract*— The Internet is omnipresent. Everyone uses the Internet for information retrieval and communication. It is part and parcel of everyday life and culture. This is also true for all levels of preschool and academic education as well as on-the-job training. Currently, providing learning objects on an electronic and personalized base is widely favored and – in the meantime – common practice.**

We present LoGen, a browser-based tool for the generation and simulation of digital, logic circuits on the gate-level via Internet. This tool is used as additional learning module for lectures and exercises at the Institute of Applied Microelectronics and Computer Engineering, University of Rostock. The feasibility of standard HTML for suchlike applications is pointed out. For the design of the tool, solely approved and well-established Internet-technologies have been used. It is especially feasible for so-called thin clients. It is shown that the Internet with its broad repertoire of basic functions and services is well suited for this kind of applications.

I. Introduction

Everybody makes use of the omnipresent Internet either as information pool or as medium for communication. In the meantime, the Internet became an integral part of our everyday life, society, and culture. This is also true for all levels of education, training, and teaching. Starting already with the preschool-age [1], to the higher academic education [2], and in-service training and education, electronic media are used to communicate all kinds of information and learning objects. Learning and teaching can be done locally on a workstation, via local area network (LAN), or using the World Wide Web (WWW) and its broad set of technologies and features that enable the ”Class Room Internet”.

Because of their direct association to research and development, academic institutions often act as precursor for new trends in education as can be seen on the high number of so-called notebook-universities [3]. Teleteaching and Distance Learning [2], ITS (*I*ntelligent *T*utor *S*ystems) [4], [5], [6], and Blended Learning [7], [1] are some example trends that are actually used to communicate content of teaching via Internet technologies. Schools, universities, and companies that do not or cannot follow suchlike trends, have to fight with regressive numbers of pupils, students, and apprentices. The use of new and up-to-date technologies for education increases the attractiveness of these institutions for new generations of learners.

Blended Learning is a superset of selected, adapted learning and teaching methods to optimize the teaching results. According to Singh [7], it is defined as ”Optimization of learning and teaching results through provision of the ‘right‘ teaching method for the ‘right‘ individual learning style to provide the ‘right‘ knowledge to the ‘right‘ person at the ‘right‘ time”. In this context, Teleteaching and Distance Learning generally focus on the distribution of information and content by means of modern telecommunication technologies across wide geographical distances and borders, for example online-lectures. As a special kind of Teleteaching, ITS provide dialogues and interfaces that communicate and interact with the learning person. ITS present the learning objects in an adapted way for the intended target group.

Blended Learning emphasizes the services and the content that is to be communicated. Here, the technology is transparent and takes a backseat – because of the manifold existing technical options in the Internet that can be utilized for suchlike learning and teaching purposes. In general, client-server architectures crystallize out. But their internal structures and realizations show different mechanisms. Examples are XML-based

(e*X*tensible *M*arkup *L*anguage) [4] or Java-based ITS [5].

Following, a tool for the generation and simulation of digital logic circuitries is presented. Its name is LoGen (*Lo*gic *Gen*erator). This tool can be used with every Internet browser. It belongs to the group of ITS. We want to show that just basic Internet technologies and mechanisms like plain HTML (*H*yper*T*ext *M*arkup *L*anguage) are sufficient for small and adapted ITS. This straightforward realization is especially suited for so-called thin clients. Section II presents the tool and its technical realization. Section III briefly introduces Hades – an existing tool – and contrasts the characteristics of LoGen and Hades. In Section IV, the current use of LoGen at our institute is discussed. Section V summarizes the results and concludes the paper.

II. LoGen - A Tool for the Generation and Simulation of Digital Circuits

LoGen is a tool for the generation and simulation of digital circuitries. It can be used within a conventional Internet-browser that belongs to the repertoire of todays personal computers and operating systems. The goal is, to easily and quickly design and test small and average sized digital circuits within the scope of our courses. Thereby, we focus on the simulation of the logical function of the circuit and comparison with truth- and state-tables. We do not simulate the physically and temporally correct behavior of the signals. LoGen serves as extension to the courses and lectures. It is a convenient and handy application to consolidate the theoretical basics.

A. Technical Background

The tool is based on a client-server architecture as sketched in Figure 1. The basic mechanism is the generation of dynamic HTML websites. Therefore, we are using PHP scripts (*P*ersonal Home Page *H*ypertext *P*reprocessor) that conform to the CGI standard (*C*ommon *G*ateway *I*nterface). Thereby, the user interacts via browser with a website. All user actions and information are forwarded to the PHP-CGI scripts via our web server. From these information, the PHP server generates a new website and transmits it as response back to the user, again using the web server. For interaction with the user and for presentation of the content, solely the options and mechanisms are used that plain HTML offers. No Flash or Java applets are applied. This demonstrates that the functional scope of plain HTML already allows for the development of complex applications.

Figure 2 sketches the principle flow within our browser-based tool. A digital circuit is displayed as an

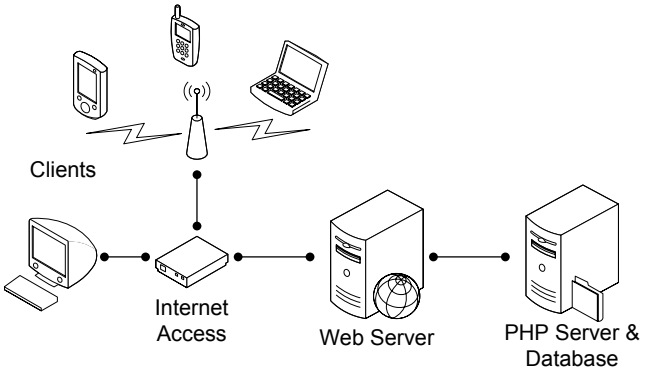

Fig. 1. Client-Server Architecture of our Tool

interactive website. The content and data of the circuit is stored in a file. Interactive means that all elements of the circuit can be referenced using hyperlinks. Every call of a hyperlink, e.g., for adding new elements to the circuit, initiates the generation of a new HTML website. Via CGI, parameters are passed to the PHP server, which generates the new webpage's content. These parameters consistently and unambiguously describe the single elements within the circuit and the circuit itself. This way, changes can be applied to the circuit currently displayed in the browser window. But no static HTML websites exist on the server. The modifiable and dynamic circuit parameters have to be stored in a file within the server's database.

B. Digital Circuits in LoGen

With LoGen, digital circuits can be designed on the level of logic gates. Single transistors are not supported. Simulation is solely done on the digital level. Thereby, the size of a circuit is not limited. Different classes of elements can be used for a digital circuit in LoGen as listed below and shown in Figure 3.

- basic logic gates like (N)AND & (N)OR for n-input gates, inverters...
- register elements like latches and flip flops (though registers can also be built using the logic gates)
- bidirectional diode elements
- wire elements, wire crossings, and wire connections
- displays and light bulbs
- static & dynamic stimuli
- some special elements & text elements

Basic logic gates, register elements, and diode elements are used to construct the functionality. Wire elements needed to connect the functional elements. Displays and bulbs are especially used for the simulation. Stimuli elements provide either static input values or alternating logic gages. When feedback loops are used

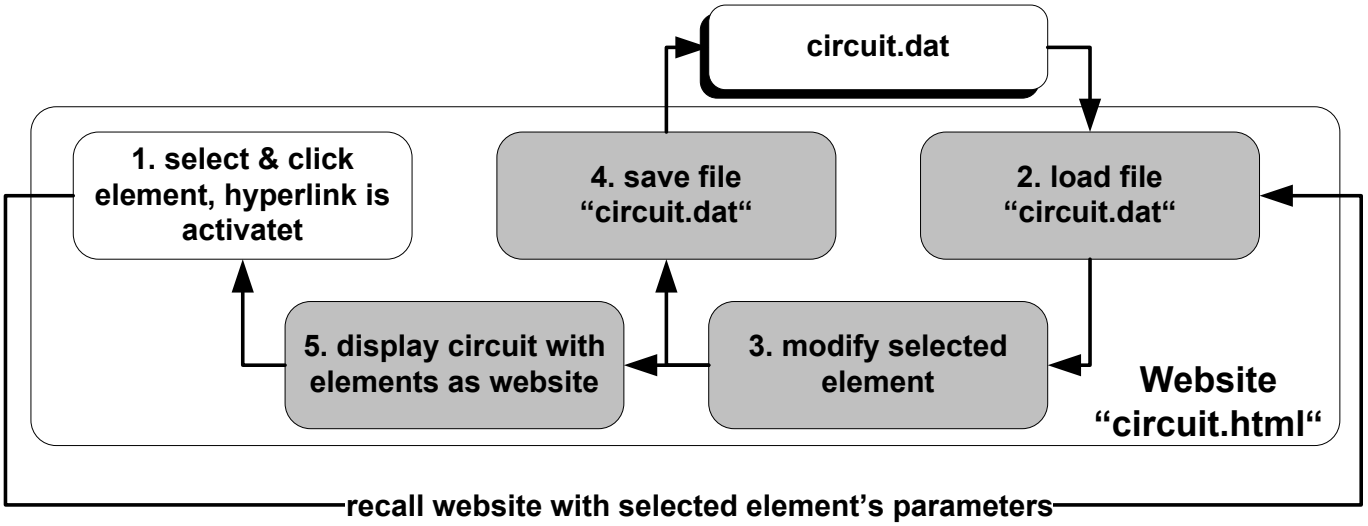

Fig. 2. Circular flow during the generation of a new circuit and refreshing the dynamic website in the browser window

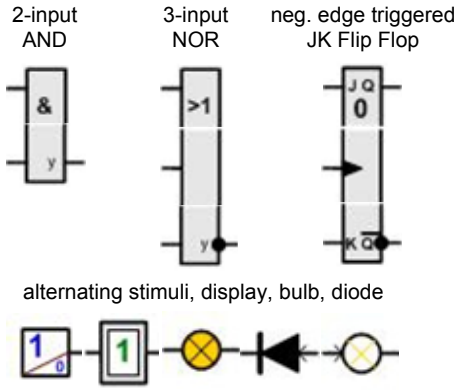

Fig. 3. Some elements of LoGen

in the circuit, e.g., within the example RS-latch in Figure 4 and Figure 5, special elements are needed for the simulation. These elements prevent the simulation algorithm (Mealy automaton) from continuously looping through the circuit by symbolically cutting the wire at a certain point.

C. How to Generate and Simulate a Design

The use of the tool consists of two main steps. At the beginning, a registered and logged-in user has to generate a new digital circuit. Alternatively, a former circuitry can be loaded from a file. As shown in Figure 4, a circuit is organized as a two-dimensional field. It is handled like a table with dynamic numbers of rows and columns. Every circuit element is characterized by a distinct position, an individual function, and an appropriate graphical symbol for the visual representation of the circuit. From the users point of view, the circuit is generated by adding and combining blocks similar to a brick set or construction kit. The symbols are ordered in categories and are intuitionally and easily graspable. They allow for an efficient generation of the circuit. Finally when generation is finished, the circuit is saved to a file.

The second step is simulation. This procedure differs from the first step in the way that every element is characterized with an additional attribute – 8its current state. Using the elements’ functions and their interconnections among each other on the structural layer, the simulation mechanism processes correct output values on the state layer for each field in the two-dimensional table. These output values depend on static or dynamic stimuli, which are generated and assigned by special functional elements attached to the inputs of the circuit. The internal simulation mechanism is based on a Mealy automaton [8], on recursive computation, and analysis of directed graphs [9]. Thereby, the static structural layer is evaluated in conjunction with the dynamic state layer. To display the circuits response and outputs, e.g., to examine or proof the functionality, special elements like light bulbs or numeric displays can be used. These special elements are also sketched in the exemplary RS-latch simulation in Figure 5.

D. Requirements to work with LoGen

There are just little requirements to run and work with LoGen. It is neither necessary to install additional software nor is high computation power required. Thus, a so-called thin client is sufficient to work with the tool. Mostly, educational and academic institutes cannot afford the latest high-performance workstations within their PC pools as well as pupils, students, and appren-

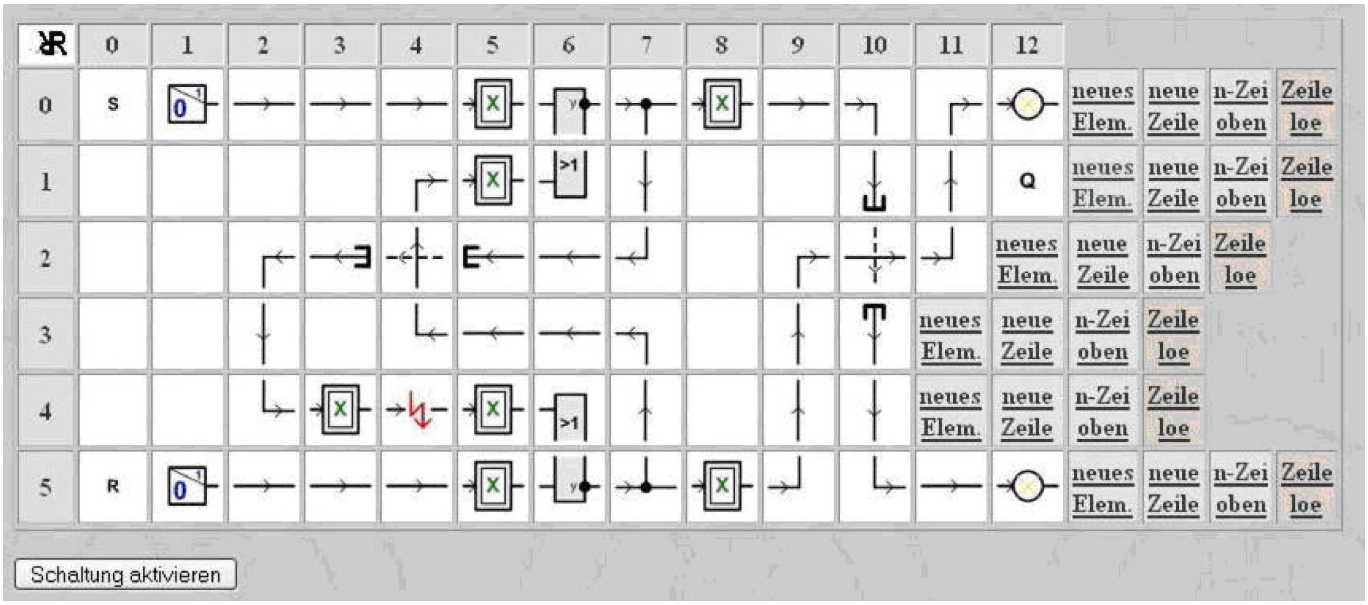

Fig. 4. User interface during generation of an exemplary RS-latch circuit

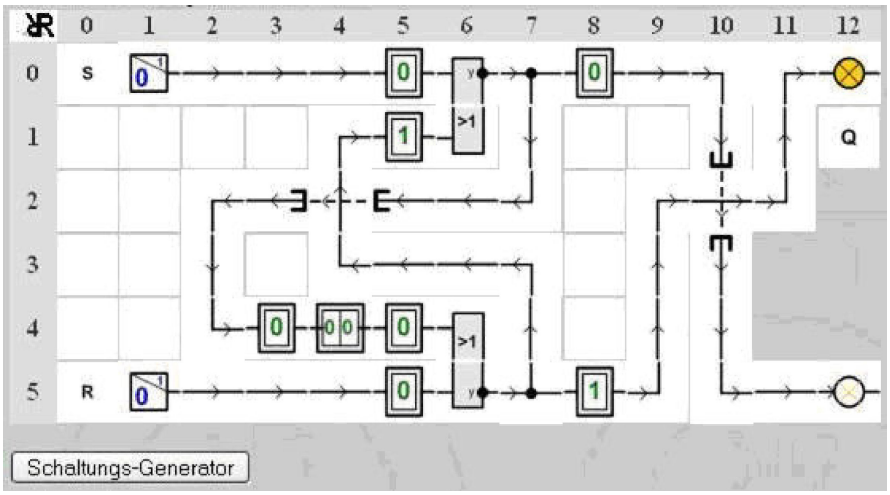

Fig. 5. User interface during simulation of the exemplary RS-latch circuit from Figure 4

tices that usually do not have the money for up-to-date hardware. Furthermore, a thin client can also be a mobile device, e.g., a laptop, a PDA, or even everyday gadgetry. The latest devices feature operating systems and access to the Internet with wireless connections. The only requirements to work with the tool are access to the Internet and a common web browser running on the system of choice. Thereby, the connection to the Internet should provide enough bandwidth to guarantee a comfortable operation and stress-free learning. Due to the dynamic generation of new HTML pages, data and information is continually exchanged. No broadband connection is required, e.g., xDSL (*D*igital *S*ubscriber *L*ine) or cable, although these types of connection to the Internet are widespread today. But also current wireless transmission media, for example, provide enough bandwidth to handle the traffic. With conventional and outdated dial-in connections, working with the tool would not be effective and just less motivating. Especially in education, the motivation of the pupils and students plays an essential role and benefits learning success!

III. Separation from existing Tools

A similar tool already exists, which also provides functionalities to generate and simulate digital circuits on the gate-level [10], [11]. This tool is called Hades and provides extensive features for the generation and simulation of circuits. Circuits can be designed on the

TABLE I
COMPARISON OF LOGEN AND HADES

Criterion	LoGen	Hades
Target Group	freshman & undergraduate students	advanced and graduate students
Purpose	Learning & Education, support Lectures and Exercises	System Simulation & Verification
Realization	HTML	Java
Structure	(thin) client - server	(rich) client - server
Requirements	Internet connection, web browser	Internet connection, special web browser, latest Java SDK
Bandwidth Demands	average	high
Interface	interaction by use of hyperlinks	sophisticated interface, multiple windows & applets
Adaptation Phase	approximately 5 minutes	training and practice required
Abstraction Level	digital logic, gate-level	gate-level, system level, HW/SW codesign
Functionality	generation, functional simulation	circuit generation, timing simulation, execution of Java applets

gate level, system level, and also in combination with software modules. A waveform viewer and other helpful features are also provided. Hades is based on the object oriented language Java. Thus, the drawback of Hades in comparison with LoGen are its requirements. The latest version of the Java Development Kit (JDK) is recommended to work with the tool. Furthermore, enough bandwidth is commended to load the applets. Not all browsers can handle Hades due to incompatibly reason between various Java releases and special browsers. Furthermore, the time to get familiar with Hades' handling is much longer than with LoGen. In the end, Hades is an excellent tool with rich features, especially for advanced users and graduate students. But this is not the focus of our tool. We target freshman, beginners, or undergraduate students. Simple functionality and quick handling are significant. LoGen is not a tool for design, development, and verification like Hades. LoGen is an additional learning module for students. Thus, plain HTML mechanisms suffice. In Table I, LoGen and Hades are compared in table form regarding various aspects.

IV. CURRENT USE OF LOGEN

LoGen is currently used at the Institute of Applied Microelectronics and Computer Engineering. Especially for the students of the lower terms, it is used to accompany the technically oriented lectures. It provides an additional learning method to consolidate the theoretical content provided during the lectures. It is also used during the exercises for practical application, to prepare and accomplish laboratory tests, and to self-evaluate the own work and ideas.

Due to its ubiquitous availability, the tool can be accessed from everywhere as long as an Internet connection exists. From within the intranet of the university (LAN/WLAN) or from outside. The students have access to the tool everywhere and everytime. Furthermore, the generated circuits are stored for each registered user and can be reloaded on demand and continued.

Currently, the number of users is limited to 20 users that can use the tool at the same time. This corresponds to the typical size of a class at the institute. But this limitation is configurable and can be adapted according to the workload and the total number of registered users.

V. CONCLUSION

We presented LoGen, a browser-based tool for the generation and simulation of digital circuits. Its client-server architecture is based on dynamic HTML pages that are generated with CGI-conform PHP scripts. The requirements to work with this tool are reduced to a minimum. Only access to the Internet and a standard web browser without any plugins are needed. Thus, it is also feasible for thin clients. The tool is used during the academic education at our institute as additional learning and teaching method. LoGen focuses on technological basics . The Where and the When is left to the students. Flexibility during learning, using modern technologies, and the focus on the technology itself are essential for the motivation of the students.

The tool can be visited and tested using the URL "http://www.elektrotechnik.uni-rostock.de/logen". A guest-login to test the simulation of the sample RS-latch exists (Login: *xxx* Password: *yyy*). A user guide and information on how to request an individual login is also provided with this URL. An individual login is necessary for the generation of new circuits.

REFERENCES

[1] Kindersite Project, "Kindersite – where children play and teachers learn," 2006.

[2] W. Schauer, "DILARC - Distance Learning and Remote Control," *Global Journal of Engineering Education*, vol. 9, no. 3, pp. 261–265, 2005.

[3] P. Deussen, W. Juling, and B. Thum, *Nukath - Die Notebook-Universitaet Karlsruhe (TH)*. Universitaetsverlag Karlsruhe, July 2004.

[4] W. Billingsley, P. Robinson, M. Ashdown, and C. Hanson, "Intelligent tutoring and supervised problem solving in the browser," in *Proceedings of the IADIS International Conference WWW/Internet 2004*, Madrid, Spain, 2004, pp. 806–811.

[5] A. Alpert, M. Singley, and P. Fairweather, "Deploying Intelligent Tutors on the Web: An Architecture and an Example," *International Journal of Artificial Intelligence in Education*, vol. 10, pp. 183–197, 1999.

[6] J. Billingsley and W. Billingsley, "The animation of simulations and tutorial clients for on-line teaching," in *Proceedings of the 15th Annual Conference for the Australasian Association for Engineering Education*, Toowoomba, Australia, 2004, pp. 532–540.

[7] H. Singh and C. Reed, "Achieving success with blended learning, centra software, white paper," 2001.

[8] B. Khoussainov and A. Nerode, *Automata Theory and its Applications*. Birkhauser, August 2001.

[9] R. Sedgewick and C. J. VanWyk, *Algorithms in C++*. Addison-Wesley Professional, December 2001.

[10] N. Hendrich, "Hades Interactive Simulation Framework," 2005, http://tams-www.informatik.uni-hamburg.de/applets/hades.

[11] ——, "A Java-based Framework for Simulation and Teaching," in *Proceedings of the 3rd European Workshop on Microelectronics Education (EWME 2000)*, Aix en Provence, France, May 2000, pp. 285–288.