

# LogiFlash – Understanding Digital Technology

## An interactive flash-based simulation tool for digital circuits

Markus Damm<sup>1</sup>, Friedrich Bauer<sup>1</sup>, Gerhard Zucker<sup>1</sup>, Klaus Waldschmidt<sup>2</sup>

<sup>1</sup>Vienna University of Technology, Institute of Computer Technology

<sup>2</sup>Goethe University Frankfurt/Main, Computer Science Department

**Key words:** *e-learning, simulation, automatic task evaluation, open source*

### Abstract:

*LogiFlash is an Adobe Flash based application to create and test digital logic circuits including logic gates, flip flops, buttons, digital displays, adders and many other components. Components are positioned and connected by pointing and clicking; the application runs without installation in any web browser that supports Adobe Flash. LogiFlash has been designed for education and allows running the final design and testing it interactively. The launch of an open source LogiFlash project is planned.*

## 1 Introduction

In both university and school level teaching, there is an ongoing trend for an increased usage of e-learning methods and tools, like animations, videos or interactive self-tests. This trend is favorable for the field of computer engineering, since visual elements like schematics, diagrams and graphs are often used for teaching, and are predestined to be brought alive by animation and simulation. LogiFlash [1, 2] is an Adobe Flash-based tool which enables instructors in the field of digital logic to enrich their lectures and online-learning material with interactively useable digital circuits and automatically evaluable exercises. The circuits are constructed using the LogiFlash-Composer, which provides the functionality of a logic simulator, and additional features like creating test pattern for exercises, defining custom components from circuits, and VHDL-export.

LogiFlash development was started at the department for Computer Science at the Goethe University, Frankfurt-Main/GER in the course of the project “Wissenswerkstatt Rechensysteme” [3], which was founded by the German ministry for Education and Science. Development continues today at the Vienna University of Technology; it is used at various universities and schools in Germany and Austria.

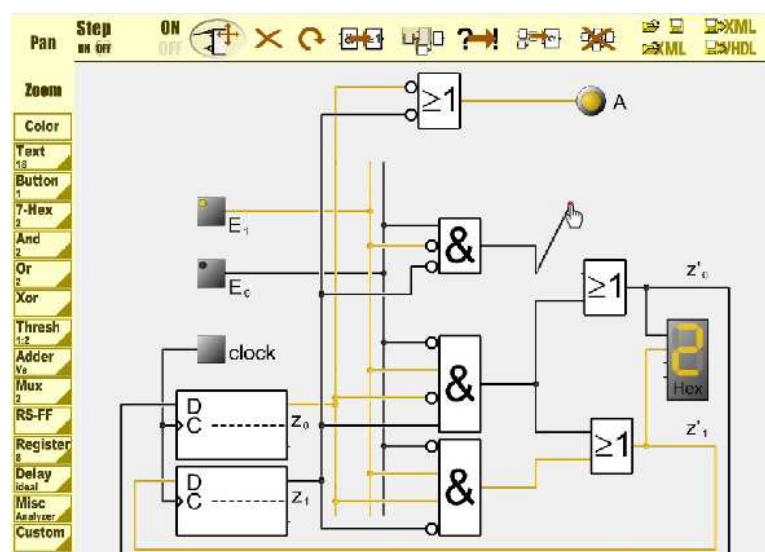


Figure 1: Screenshot of the LogiFlash Composer

## 2 Features

The LogiFlash Composer (see Figure 1) offers the functionality of a Logic Simulator with schematic entry: The user can add various components (e.g. logic gates, multiplexers, flip-flops, in- and output devices) to the workbench, and connect them with wires. The resulting circuit then can be simulated, where the simulation result is visualized by illuminating the wires according to the logic values they carry, i.e. black for 0 and yellow for 1. It is also possible to construct custom gates out of existing circuits, where the resulting gate offers a “zoom in“-feature to inspect the original design. Also, text can be added. Circuits can be stored in an XML format, and can also be exported to VHDL.

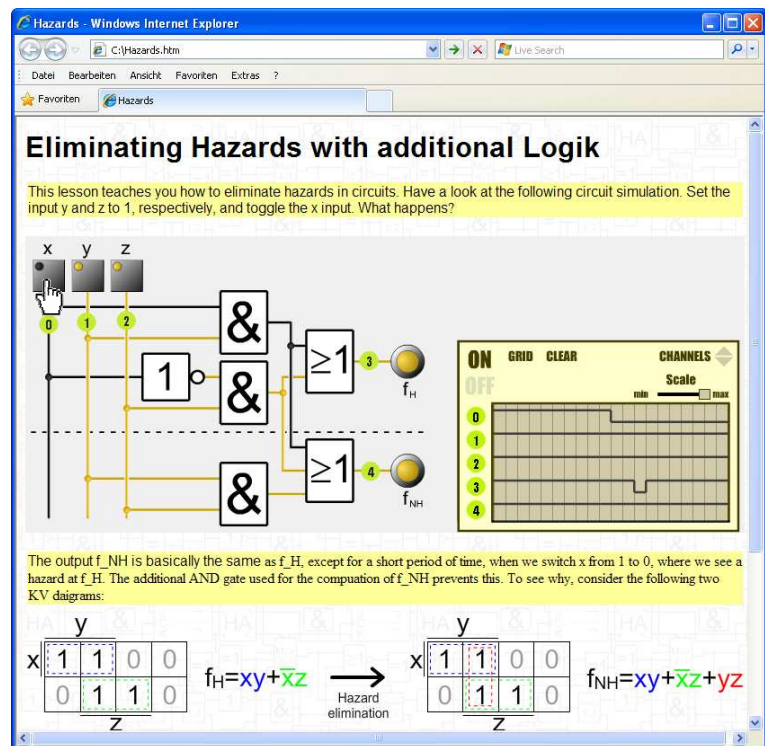


Figure 2: The LogiFlash Viewer embedded within a HTML page

An emphasis lies on the visualization of delays. All gates have a (uniform) delay long enough to see how signals propagate through circuits, and also to observe effects like races and hazards. Furthermore, dedicated delay elements can be used to add further delay, which may also react differently on rising and falling edges. For detailed analysis, a logic analyzer is available (see Figure 2); alternatively, the simulation can be run stepwise.

A distinctive feature of LogiFlash is to give teachers the ability to create automatically evaluable exercises, that is, the solution of students are checked automatically for correctness. The standard workflow is simple: A sample solution is created, i.e. a working circuit for the task at hand. From this sample solution, a test pattern, i.e. a series of inputs and outputs, can be created. In case of stateless circuits, this would be simply the corresponding function table, which can be computed automatically. But also custom test patterns can be created, e.g. for stateful circuits.

The students' task then is to create a circuit which matches the test pattern. For example, they have to work with a modified version of the sample solution circuits, where some gates might be missing or are exchanged. The correctness of the solution then can be verified by applying the inputs of the test pattern to the circuit and comparing them to the output values stored within the test pattern.

Since LogiFlash is an Adobe Flash application, it can be used in HTML documents, PowerPoint slides, and all e-learning standards and applications supporting Flash, e.g. SCORM modules. For many applications, there is no need for the full menu set of the LogiFlash Composer, e.g. when an author wants to display and simulate a circuit within a HTML document with no further user interaction apart from manipulating the inputs. For such cases, it is possible to use the LogiFlash Viewer, which can be customized to use no menus (for simple display & simulation) or a reduced menu set for exercises or the step mode. Also, the LogiFlash Viewer offers a simple Flash API, such that it can be embedded within other flash animations which then can read the outputs of LogiFlash circuits or manipulate their inputs.

### 3 Use cases

There are several ways to use LogiFlash for teaching. It is common in teaching of basic digital logic to include pictures of circuits; e.g. in PowerPoint slides for lectures. With the LogiFlash Viewer, these circuits can be brought to life, and the lecturer can demonstrate how the circuit actually works. If the lecture is accompanied by online learning material, the same circuit simulation can be included there. See Figure 2 for an example.

The actual didactic goals of these animated circuit simulations can be divided into two categories: The first category encompasses the understanding functional properties of logical circuits, e.g.

- Combinational and sequential circuits
- Two-level and multilevel circuits
- Multiplexer-based implementations
- Commonly used circuit designs

The second category deals with non-functional properties:

- Propagation delay
- Hazards (see Figure 2)
- Race conditions
- Synchronous vs. asynchronous circuits

For both categories, LogiFlash is a suitable didactic tool for interactive visualization on the circuit level. However, there are of course other forms of visualization with respect to digital logic, like KV diagrams or state graphs for finite automata. By embedding the LogiFlash Viewer within other Flash animations, it is possible to make a lively connection between these theoretical concepts and the corresponding circuit realisations.

Figure 3 shows an example: A flash animation simulates traffic on an intersection with traffic light and induction loops. The logic to control the traffic lights is actually realised as a LogiFlash circuit, which is simulated with an embedded LogiFlash Viewer. The inputs of the circuit are controlled by the main animation via the API offered by the LogiFlash Viewer, and via this API it also reads back the results of the circuit simulation, which is used to control the traffic lights. Additionally, the corresponding state graph is shown where the actual state and the next state transition are highlighted.

Automatically evaluable exercises can be used in different ways. The type of task can range from finding errors within existing circuits to building new circuits from scratch (see Figure 4). The tasks can be self-test within learning objects up to tasks relevant for grading. In the latter case, this can be done either online or offline.

Finally, the LogiFlash Composer provides students with an easy to use digital logic simulator to explore the world of digital logic by their self.

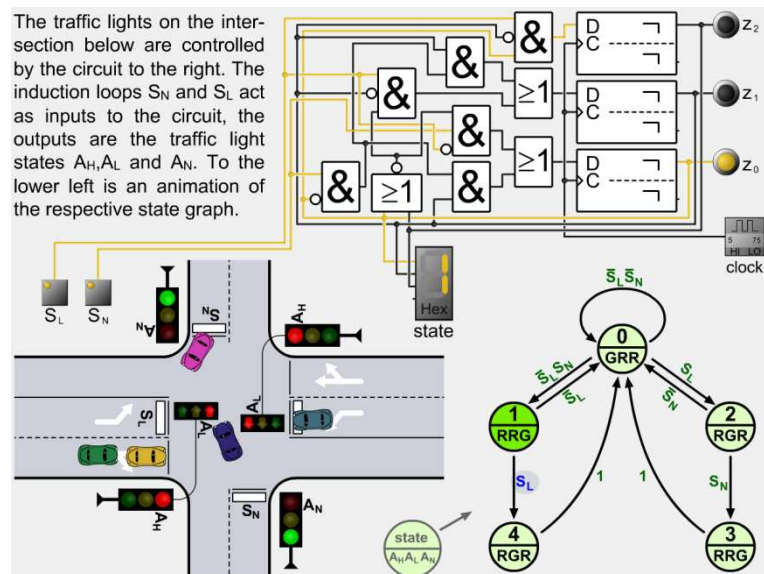


Figure 3: A LogiFlash circuit embedded within another Flash animation

## 4 Practical experiences

LogiFlash has been used at the Goethe University, Frankfurt-Main/GER for teaching of digital logic to computer science students in the second semester. LogiFlash simulations were embedded into the lecture slides and also available to the students online. An experience made was that the lecturer should not demonstrate every detail of every circuit simulation which was embedded, but to concentrate on the most important circuits and aspects, since all the circuit simulations were available to the students online anyway.

Also, a supervised hands-on lab (with about 8 students groups) using LogiFlash was held as part of the lecture, where the student's results actually contributed to the grade. While most tasks were automatically evaluable construction tasks (see Figure 4 for a translated version of one of the tasks), some tasks involved also timing behaviour, like manipulating delays to make an asynchronous circuit work. The latter tasks actually needed the supervisor to check the solution, while in the other cases, he just provided help and noted the results.

At the Vienna University of Technology, LogiFlash is used for a curriculum-relevant online lab within a SCORM module [4] that is embedded into the e-learning platform tuwel [5]. The task was to construct a sequential circuit from a text specification. The exact task that a student had to work on was chosen from a pool of tasks depending on the learner id. This application got an e-learning award from the Vienna University of Technology's e-learning centre [5].

## 5 Open source relaunch

Despite its advantages, LogiFlash has also some shortcomings. For example, the user interface is available in German only, which prevents international distribution. Also, only DIN standard symbols are available, but not international more common ANSI symbols. Since two-valued logic is used, some common circuit designs which use components like tri-state gates cannot be realised. Furthermore, LogiFlash uses ActionScript 1.0, which limits its use within Flash animations using ActionScript 3.0, since the LogiFlash API is not available in this case.

Since there is no more funding available, transforming LogiFlash to an open source project is the most obvious choice for future development. Currently, in the course of a Bachelor thesis at the Vienna University of Technology, a LogiFlash kernel using ActionScript 3.0 is implemented, which will be the base for the open source project. Based on this kernel, Custom LogiFlash applications can be built, e.g. internationalized LogiFlash Composers, or different LogiFlash Viewer versions for certain application fields, e.g. to be used within SCORM modules. For the time being, LogiFlash has been made open source and can be accessed at [6].

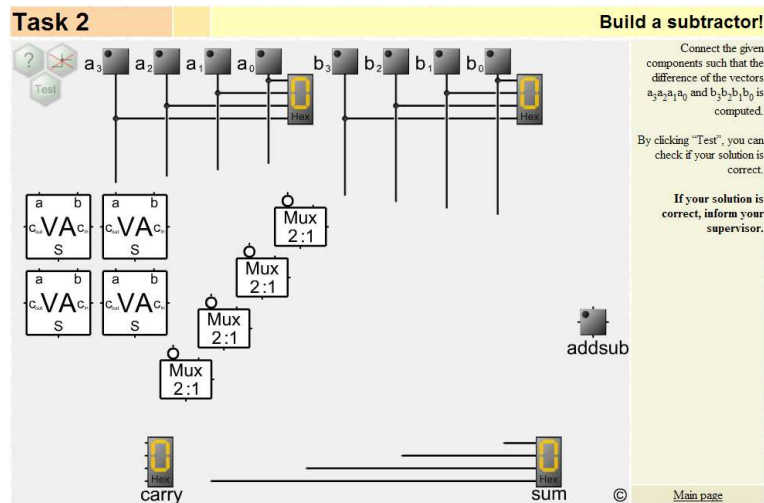


Figure 4: Example of a LogiFlash task

## 6 Conclusion

This paper presented the features and usages of the e-learning focussed digital logic simulator LogiFlash, which offers support for automatically evaluable exercises. It was shown that the chosen Adobe Flash platform allows for unique e-learning applications linking theoretical concepts to circuit implementations. We hope this paper sparks interest among those engaged in e-learning for computer engineering to participate in the LogiFlash open-source project.

### References:

- [1] Damm, M., Klauer, B. & Waldschmidt, K. LogiFlash - A Flash-based Logic-Simulator for educational Purposes. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003 (pp. 748-750).
- [2] LogiFlash Examples and Download: <http://www.ti.cs.uni-frankfurt.de/wwr/>,  
<http://pii-workshop.hsu-hh.de/LogiFlash/index.html>
- [3] Kornelsen, L., Lucke, U., Tavangarian, D., Voigt, D. & Waldhauer, M. Inhalte und Ergebnisse des Verbundprojekts Wissenswerkstatt Rechensysteme (WWR), GI Softwaretechnik-Trends 24/01, February 2004.
- [4] Damm, M., Bauer, F., Zucker, G. Solving Digital Logic Assignments with Automatic Verification in SCORM Modules, accepted at the ICL 2009
- [5] <http://elearning.tuwien.ac.at/index.php?id=442>
- [6] <http://logiflash.sourceforge.net/>

### Author(s):

Markus Damm, Dipl. Math.  
Friedrich Bauer, Dr. techn.  
Gerhard Zucker, Dr. techn.  
Vienna University of Technology,  
Institute of Computer Technology  
Gußhausstraße 27-29/384  
1040 Vienna  
{ daml|bauer|zucker }@ict.tuwien.ac.at  
Markus Damm, Dipl. Math.

Klaus Waldschmidt, Prof.Dr.  
Goethe University Frankfurt am Main  
Robert-Mayer-Straße 11-15  
D-60325 Frankfurt am Main  
waldsch@ti.cs.uni-frankfurt.de