Gift – An Integrated Development and Training System for Finite State Machine Based Approaches

https://doi.org/10.3991/ijoe.v13i08.7387

Karsten Henke(✉), Tobias Fäth, René Hutschenreuter, Heinz-Dietrich Wuttke
Ilmenau University of Technology, Germany
Karsten.Henke@tu-ilmenau.de

Abstract—At the Ilmenau University of Technology's "Integrated Communication Systems" Department a main teaching concept deals with the design of digital control systems. Different lectures from the 1st to the 8th semester are using Finite State Machines (FSM) as a specification technique to realize different design tasks. During undergraduate studies the basics of Finite State Machines and their usage within the design of digital control systems are taught. To conceptualize more complex digital systems, as required in higher courses, it is necessary to use powerful toolsets. One example of such a toolset is the GIFT (Graphical Interactive Finite State Machine Toolset) system, developed by the Integrated Communications System Group at the Ilmenau University of Technology. With this toolset we want to extent our remote lab GOLDi and implement new techniques for a web-based development system for Finite State Machines.

Keywords—control engineering education, web-based design tools, and remote laboratories.

1 Introduction

With our hybrid online lab GOLDi (Grid of Online Lab Devices Ilmenau), described in several papers [1]-[6], we support the design process of digital control systems, which usually consists of the conceptual formulation and the design of the control algorithm to finally achieve a validated control (see Fig. 1).

For the functional description, we offer different specification techniques by using non-commercial development tools for various web-based control units in the remote lab:

- a Finite State Machine (FSM) based design on the basis of digital automata - executed within a client-side FSM interpreter
- a software-oriented design in C or assembler executed on microcontrollers
- a hardware-oriented design in hardware description languages or schematic block design by using FPGA's.

Simulation and visual prototyping help to identify functional errors before starting practical work on real physical systems (the electro-mechanical models) in the lab room with one of the selected control units.

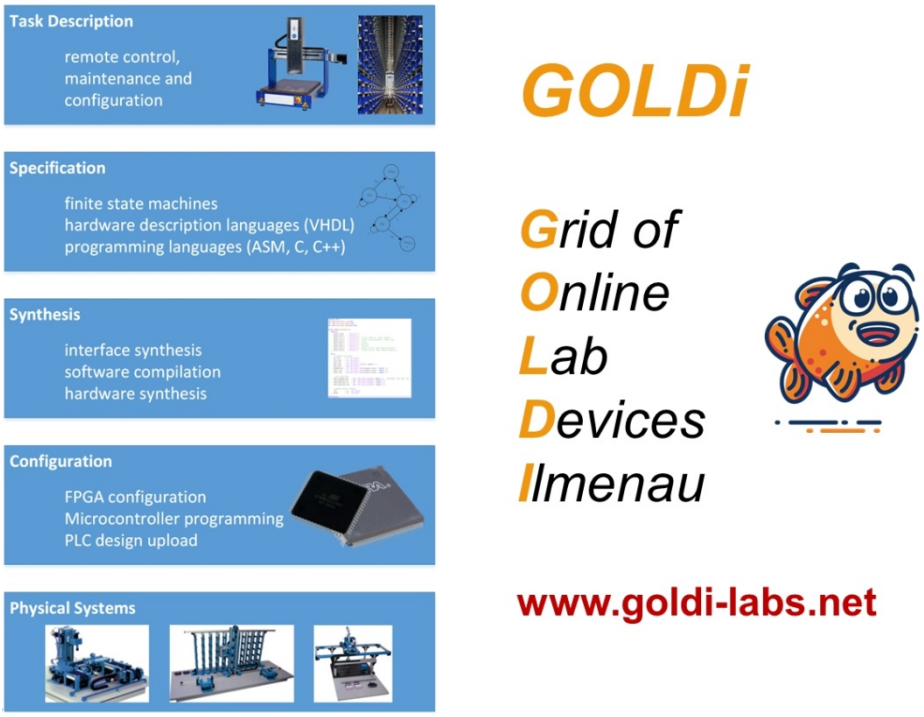

Fig. 1. GOLDi Design flow

While several professional or semi-professional design tools are available for software and hardware oriented approaches, the FSM based design is only partially supported. That's why we enhanced the functionality of our GOLDi system by the GIFT (Graphical Interactive Finite State Machine Toolset) system which will be described in detail within this contribution. The GIFT system is an integral part of the GOLDi remote lab infrastructure. The first conceptual ideas for this GIFT system were presented during REV2016 [7].

In the following, we will demonstrate how students can use the GIFT system to:

- specify the given task by using Finite State Machines
- simulate their design within the GIFT system to achieve a faultless solution and
- export the generated next-state and output equations to the GOLDi remote lab infrastructure to test their solution under real environmental conditions with electro-mechanical hardware models.

2 GIFT within the GOLDi Remote Lab Infrastructure

Based on the flexible grid structure of the GOLDi system an experiment consists of two components: on the one hand there are various control units (e.g. FSM, microcontroller, FPGA). On the other hand, there are the electro-mechanical physical systems (e.g. elevator, 3-axis portal crane or warehouse models). A detailed description of the whole GOLDi architecture as well as different working modes can be found in [1]-[4].

As shown in Fig. 2, the GIFT system is part of the toolchain which is executed at client side within a web-browser. Any client side tools will be downloaded from the GOLDi cloud [12]. Two terms are used to classify the client side tools:

- Design Environment (GIFT, Atmel Studio, Quartus Prime)
- Execution Environment (ECP- Experiment Control Panel).

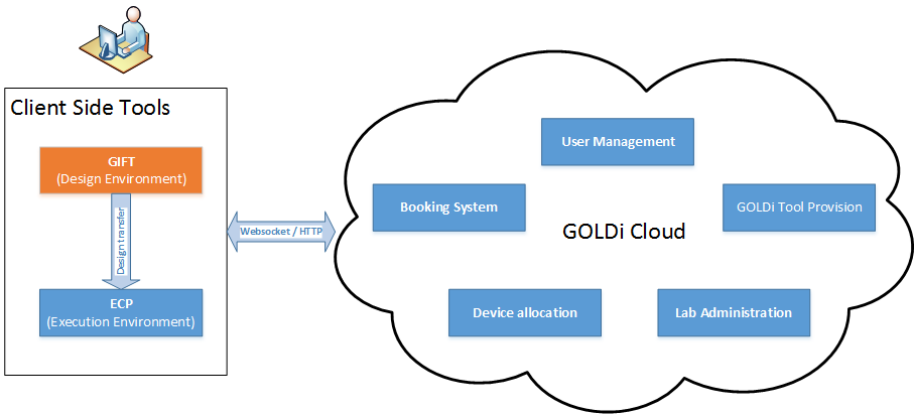

Fig. 2. GIFT within the GOLDi- Architecture

2.1 Design Environment

The Design Environment is used to specify the user developed design by using a specification technique determined by the control unit. For FSM based designs, this Design Environment is the GIFT system which provides methods for:

- general FSM administration
- FSM input as automaton graph or transition table
- handling of transition conditions and coding of states
- simulation of single and parallel automata
- generation of the next-state (z) and output (y) equations
- export of z/y equations to the ECP of the GOLDi system
- generation equations for D and JK flip-flops.

Other examples for Design Environments are Atmel Studio [8] in case of microcontroller based control units for a software-oriented implementation or Quartus

Prime from Altera [9] in case of FPGA based control units for a more hardware-oriented design. The Design Environment is not limited to specification - it provides also support for syntax check of the design in the specification language as well as automatic tools for verification and validation. An important feature is also the possibility to generate output data (e.g. GIFT export data or Atmel *.hex, Altera *.pof binary files), which can finally be loaded into the Execution Environment.

2.2 Execution Environment

The Execution Environment is used to execute the user design within the GOLDi architecture. This is done by using the Experiment Control Panel (ECP). The ECP supports any of the available control units within the GOLDi architecture and is dynamically configured on startup to fit the user's choice in control unit/electro-mechanical physical system.

By using the ECP [7], students can:

- upload the synthesized/compiled designs to the corresponding control units (e.g. microcontroller, FPGA) in the lab room
- import designs previously exported from the GIFT and execute them in the ECP's integrated FSM interpreter

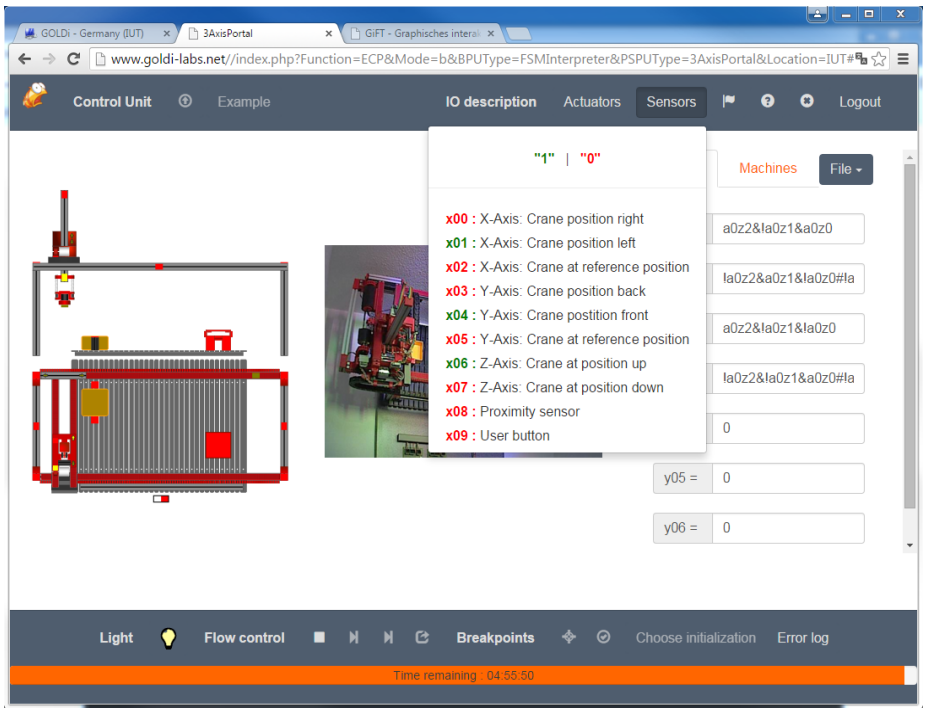

Fig. 3. I/O monitor (e.g. sensor signals) of the ECP for the 3-axis portal crane

- . watch the experiment by manipulating environmental variables inside an I/O monitor (Fig. 3 shows an example to monitor the sensor signals of the 3-axis portal) or by observing the control of the physical system directly via a webcam
- handle the experiment (e.g. start, stop, reset)
- use the interactive debugging features (break on sensor/actuator changes or special conditions)
- single step processing, by pausing the execution on every sensor/actuator change
- change environmental variables if necessary
- choose an individual initial situation for the experiment by manipulating the physical system via mouse or keypad or by choosing a pre-defined initial state.

3 FSM Based GOLDi Design Flow

By using Finite State Machines as specification technique to realize various control tasks, the graphical interactive FSM toolset GIFT can be used directly within the GOLDi design flow (see 0).

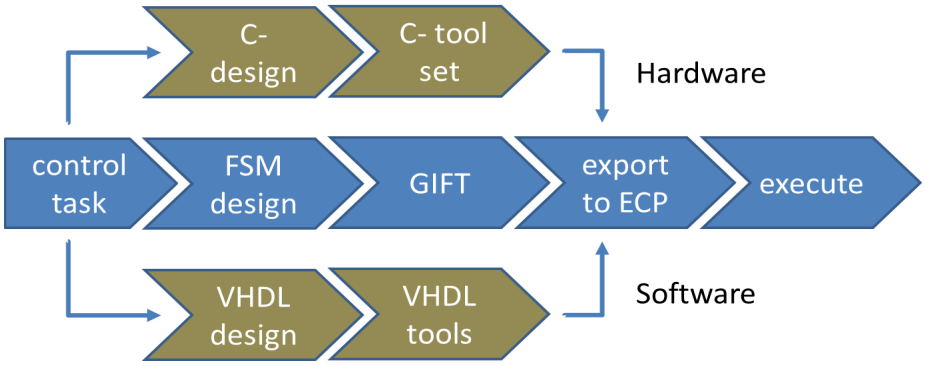

Fig. 4. GIFT integrated GOLDi design process

3.1 Control Task Example

As an example, to introduce Finite State Machines to students in the first semester, we will discuss a design task by using the electro-mechanical model "3-axis portal crane" from the GOLDi remote lab infrastructure:
"On one spindle of a 3-axis portal crane, a tool carriage can be moved to the right and to the left. Limit switches provide input information on the left end position (x_l) as well as the right end position (x_r) of the tool carriage (x_r, x_l).

The motion can be controlled via the output variables (y_l, y_r) between

- *motion to the left ($y_l = 1$, $y_r = 0$),*
- *motion to the right ($y_l = 0$, $y_r = 1$) and*
- *stop ($y_l = y_r = 0$).*

An additional input variable x_s signalizes

- *stop motion (x_s = 0) or*
- *movement (x_s = 1) to the left or right.*

After a possible break, the movement in the original direction should be continued."

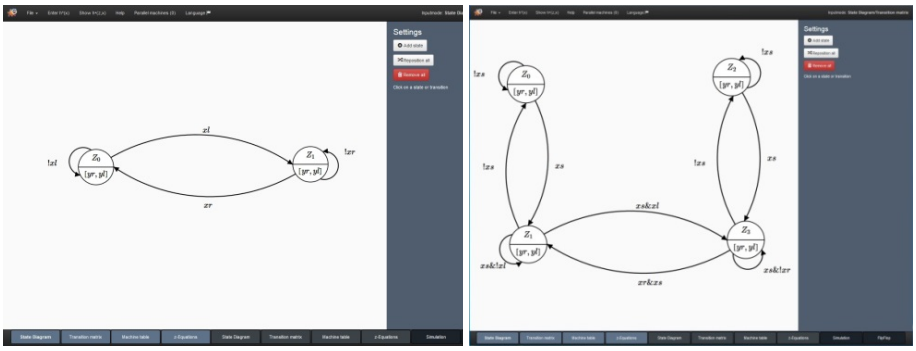

Fig. 5. Mealy (left) and Moore automata (right) for the spindle control task

3.2 GIFT FSM Design

In the following we demonstrate how students can specify their design based upon the given task by using Finite State Machines, simulate their design within the GIFT system to achieve a faultless solution step by step and can finally export the generated next-state and output equations to the ECP within the GOLDi remote lab infrastructure.

Design of the Automaton Graph Such kinds of tasks are solved by developing a formal description of the control algorithm on the basis of an automaton graph and the corresponding Boolean equations. Fig. 5 shows two possibilities of corresponding automaton graphs by using the graph editor of the GIFT system:

- a Mealy automaton graph with two states as well as
- a Moore automaton graph with four states.

Simulation of the Design The student can proceed to the simulation process using graphical controls. That leads to appropriate state transitions caused by the changed set of input variables. Via waveform simulation (see Fig. 6) the temporal sequence of input and output variables and the internal state variables (coding the states) of partial automata can be shown.

3.3 Export to ECP

Most of the students are highly motivated in realizing such design tasks in a real laboratory because they can see the results of their design immediately. That's why the GIFT system offers the possibility to export the generated next-state and output equations directly to the ECP of the GOLDi system (see Fig. 7).

Because the design is typically made with user-defined variables (e.g. x_l, y_r) the student must adapt them to the real sensor and actuator interface notations (e.g. x_r must be replaced by x_{00} of the 3-Axis Portal sensor signals, which means "X-Axis: Crane position right"). This can be done interactively for each input and output variable within the ECP (see Fig. 8).

3.4 Design Execution

After uploading the next-state and output functions from GIFT or inserting these equations manually to the ECP, students are able to start the lab procedure and watch the behavior of their implemented design (see Fig. 3).

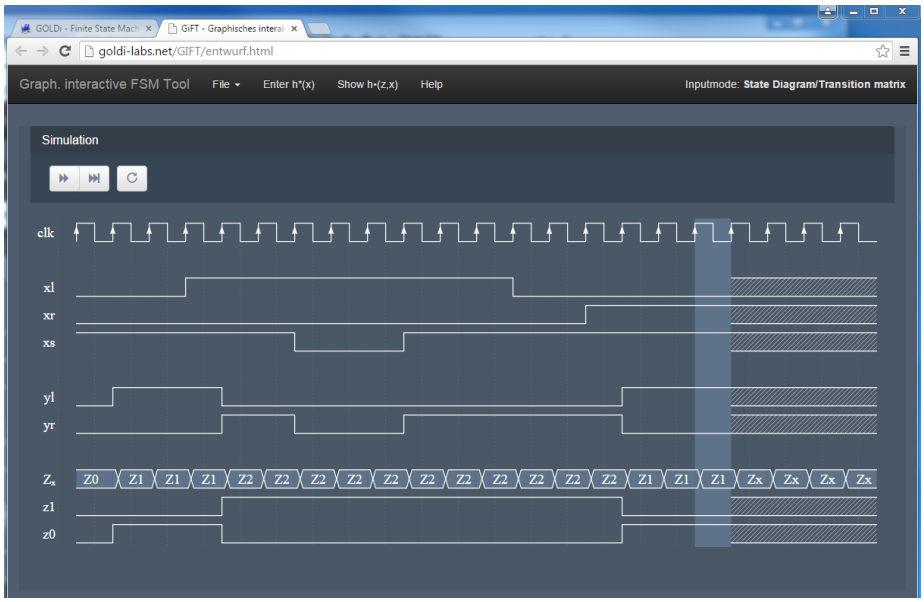

Fig. 6. Simulation tool of the GIFT system

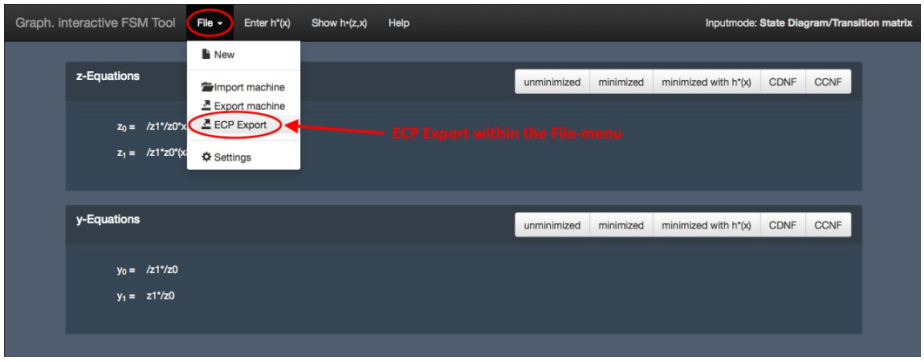

Fig. 7. GIFT: ECP export functionality

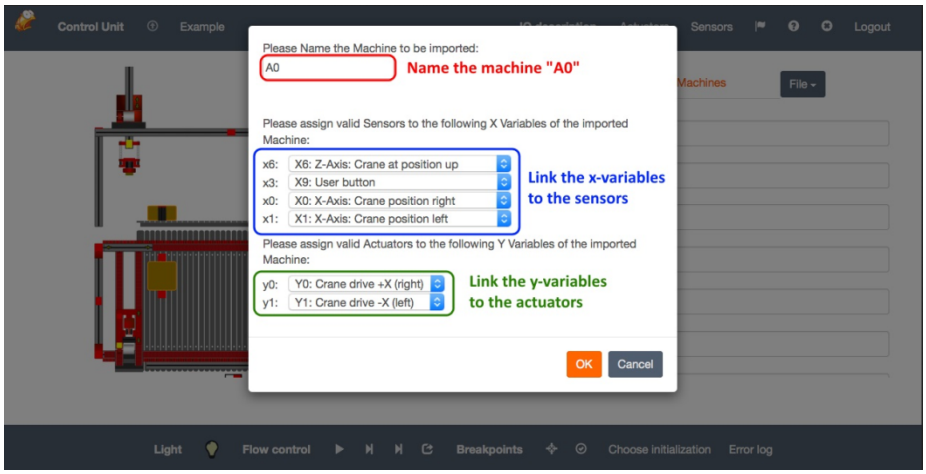

Fig. 8. ECP: GIFT import functionality

In this case, the electro-mechanical model will be controlled via the Internet "from a distance" through the interpreter, running inside the student's client PC. No additional control units in the remote lab are necessary. In this case, only the input and output signals between the virtual control and the physical system are transferred via Internet.

In case of a selected virtual physical system the real physical system is represented by a simulation component within the ECP running on the client machine. No Internet connectivity to the laboratory and the physical systems in the remote lab is necessary for this kind of experiment. This mode is especially suitable for face-to-face lectures in a classroom where each student can work with the same virtual physical system simultaneously as well as in situations in which a stable network connection cannot be guaranteed (e.g. during travel or while using mobile networks).

4 Further Applications of the GIFT System

In this chapter, further applications of the GIFT tool beside the standard GOLDi development process mentioned above will be explained. These applications are used alongside our teaching models at Ilmenau University of Technology and have been proven to be successful in teaching in depth knowledge of techniques to achieve flawless designs based of Finite State Machines.

4.1 Teaching by Demonstration

Besides teaching the basics of FSM's the GIFT system will be used in undergraduate lessons to demonstrate different, unexpected side effects which can occur when a FSM based specification is incomplete and/or contains contradictions. The GIFT

system can demonstrate the resulting faulty behavior of the design task and gives information how to avoid them.

Due to the intuitive approach for the design of automata graphs it could be possible that students

- forget to specify all next-state conditions which lead to an incomplete design or
- specify inconsistent next-state conditions which lead to a contradictory design.

In both cases it is not recommended to realize such faulty designs because the behavior of the implemented design is normally completely different as the desired behavior. Fig. 9 (left) gives an example for a faulty Moore automaton of the spindle design task with contradictions in states Z_1 and Z_3. The GIFT system automatically indicates an incorrect design by highlighting the faulty state transitions in red and calculates the behavior of the resulting design based on the faulty next-state functions (Fig. 9, right). The correct Moore automaton graph is shown in Fig. 5, right.

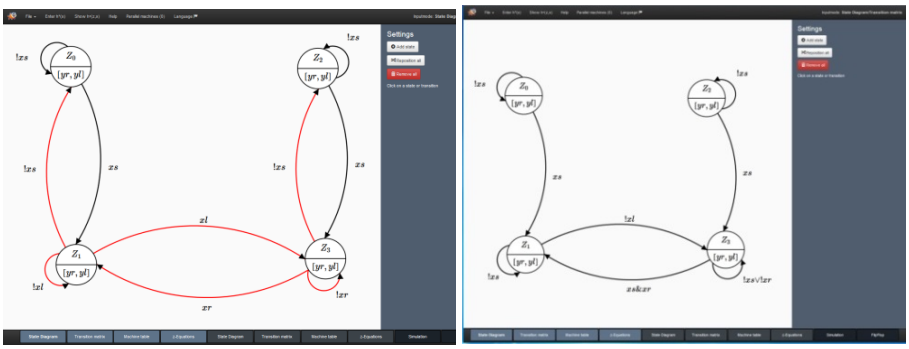

Fig. 9. GIFT: Incorrect FSM design (left) and resulting automaton graph (right) for the given spindle control task

The idea is to stop the motion during the "motion to left" (state Z_1) or "motion to right" (state Z_3) to switch to the two "stop" states Z_0 or Z_2 by deactivating the x_s variable. The motion to left or right will be continued by activating x_s again. Due to the intuitive design, students often ignore, that it is only possible to change the motion direction (between Z_1 and Z_3 or vice versa) or stay in these states when x_s is activated.

This will otherwise result in the mentioned contradictions. Although the resulting design can be activated to drive to the left or right direction from the two "stop" states Z_0 or Z_2 it is never possible to stop the motion again (see Fig. 9, below). In the worst case the motion will not be stopped in Z_3 if x_s is deactivated and the right end position is reached – which finally could damage the electro-mechanical model (or the hardware model in the f2f laboratory without any further protection mechanisms).

If students decide to ignore the automatic design checks of the GIFT system and export faulty next-state and output functions to the ECP of the GOLDi system and start the execution, the implemented protection unit will finally terminate the execution to avoid any damages of the electro-mechanical hardware models (which is shown in Figure 10).

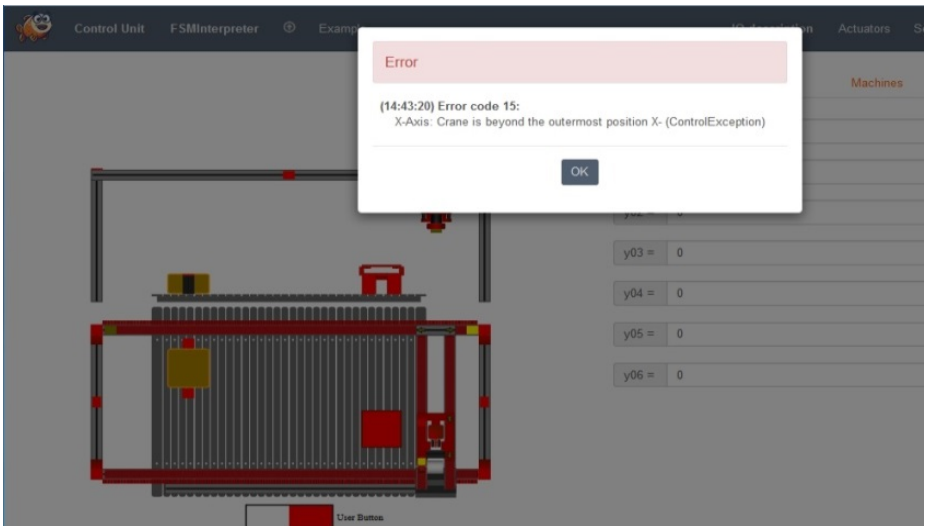

Fig. 10.ECP: Termination of the design execution due to design errors

4.2 GIFT as Interactive Training System

Complementing the theoretical knowledge taught during lectures, students can use the GIFT system as an interactive training system to deepen their knowledge in specification techniques or for exam preparation. For that, students will get predefined automata designs:

- to derive the next-state and output functions and
- to check the given automaton regarding completeness and contradictions
- and can compare their solutions with the calculations of the GIFT system.

4.3 Preparation for Hands-on Lab Sessions

In preparation of the hands-on laboratories students can enter and simulate their FSM design for the given lab task. The GIFT assisted preparation process for hands-on lab sessions is shown in Fig. 11. In contrast to the automated GIFT integration into the GOLDi infrastructure and the execution of the design in the ECP, for hands-on lab sessions the students must realize their design by manually connecting integrated circuits (e.g. AND/NAND, OR/NOR Gates, D/JK flip- flops) by wires.

Fig. 11. GIFT assisted preparation for hands-on lab sessions

Equations for D and JK flip-flops (Fig. 12) as well as output equations can be generated by the GIFT system. During the hands-on laboratory students will build up a sequential circuit on the basis of these next-state and output functions to control simple technical facilities. While the results achieved within the GOLDi infrastructure are self-assessed, in hands-on lab sessions this is done by a teacher/tutor (Fig. 12).

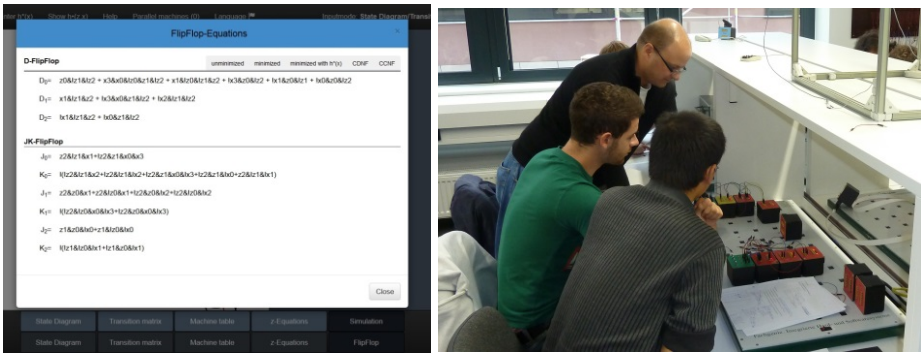

Fig. 12. GIFT: Export of minimized D Flip-Flop and JK Flip-Flop equations (left), used in hands-on lab session (right)

4.4 Design of Parallel Automata

Beside the design of single automata, students can use the GIFT system to design parallel automata. In principle, there are two possibilities:

- the automatic decomposition of a single automaton in several parallel automata or
- the intuitive design of a set of single automata which will be interact finally as a parallel system.

Automatic Decomposition of a Single Automaton into Parallel Automata This mode will be used mainly for teaching purposes to demonstrate different aspects for the design of parallel automata. Based on the design of a single automaton, students can choose the number of parallel automata and the state coding of the parallel automata. Fig. 13 gives an example for a single automaton for a water level control task.

Although GIFT detects contradictions in two states (Z_2 and Z_3, marked in red), it is possible to use this design because these contradictions are part of the "don't care"-conditions of the water level system.

The water level control task (see visual model in Fig. 13) in detail:

"The water level control experiment consists of a tank, two pumps, four water level sensors and a consumer represented by a drain valve at the bottom of the tank. The pumps are used to raise the water level and the drain valve is used to consume the water in the tank. The task is to control the pumps in a way that the water level remains between the lowest and the highest. If only one pump is active, they should be alternatively be used."

The number of active pumps as well as the alternation is described in the specification scheme (Fig. 13, right).

One example for decomposition into two parallel automata is shown in Fig. 14 and Fig. 15. The first parallel automaton (Fig. 14) is responsible for the number of active pumps:

- Z_0: both pumps are inactive
- Z_1: one pump is active – depending the actual state of the second parallel automaton
- Z_2: both pumps are active

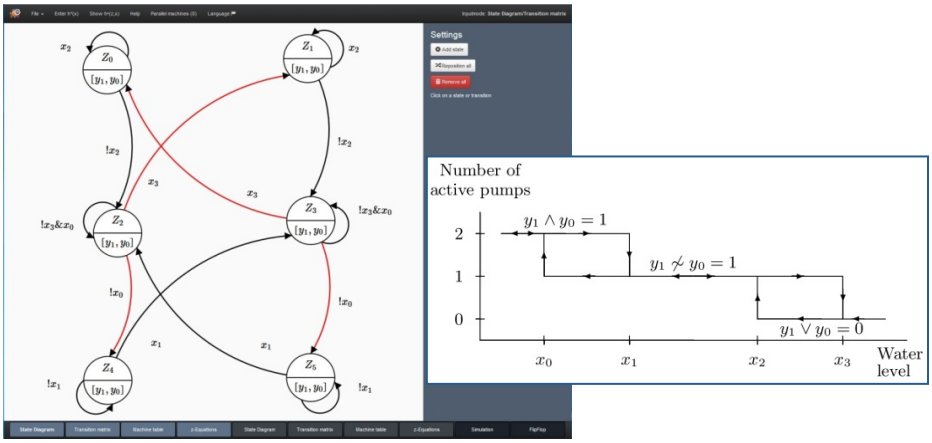

Fig. 13.GIFT: Automaton graph for the water level control task (left) and the specification scheme (right)

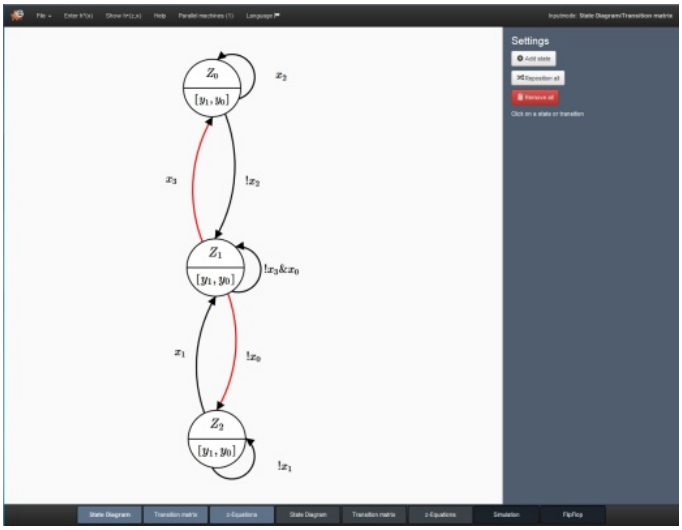

Fig. 14.GIFT: Automaton with 3 states

The second parallel automaton (Fig. 15) is responsible to switch between the two pumps for an alternative use if only one pump is active according to the specification scheme, shown in Fig. 13.

This approach of decomposition is often used in lectures and seminars to discuss special aspects with the students on the fly. To use the ECP execution for a number of students it is necessary to switch to the "virtual mode" – shown in Fig. 16

Intuitive Design of Parallel Automata In special courses the design of complex digital control systems will be taught to students of the 8th semester in form of a so called "project class". The students have to solve a complex design task and document all the steps of their work in great detail. During the development process, a collection of design variants will be created; serving as case studies and provided via the Internet. Examples for such design tasks are amongst others controls for technical facilities like elevators, production cells and high storage warehouse systems. Solutions of such controls are available e.g. as a set of parallel automata (designed with the GIFT system) within the GOLDi remote lab infrastructure. Fig. 17 gives an impression of the execution of a control task for the Production Cell – realized with 12 parallel automata.

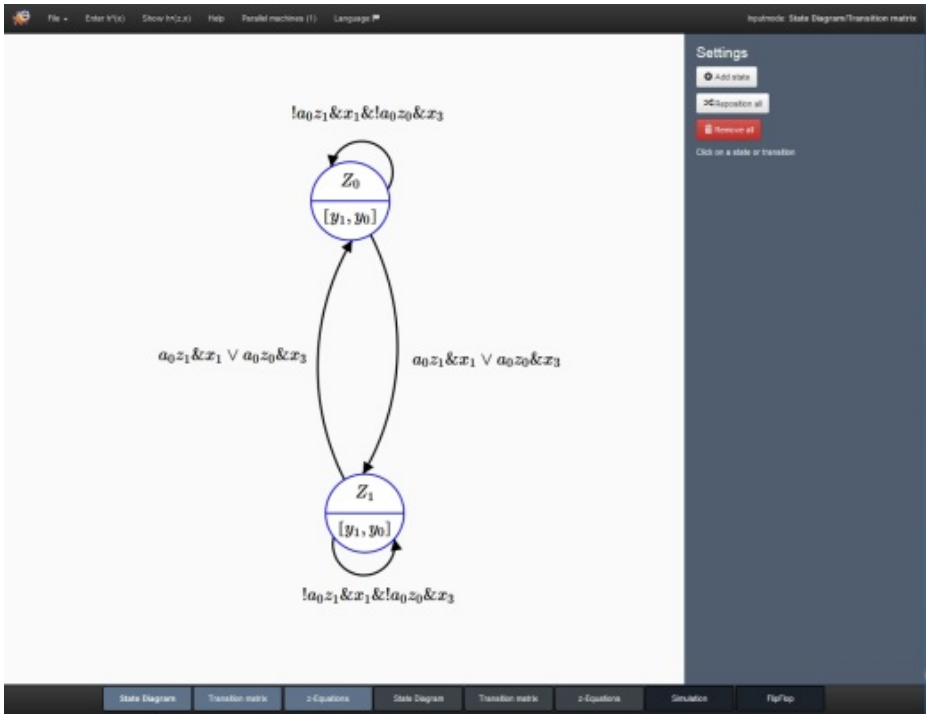

Fig. 15.GIFT: Automaton with 2 states

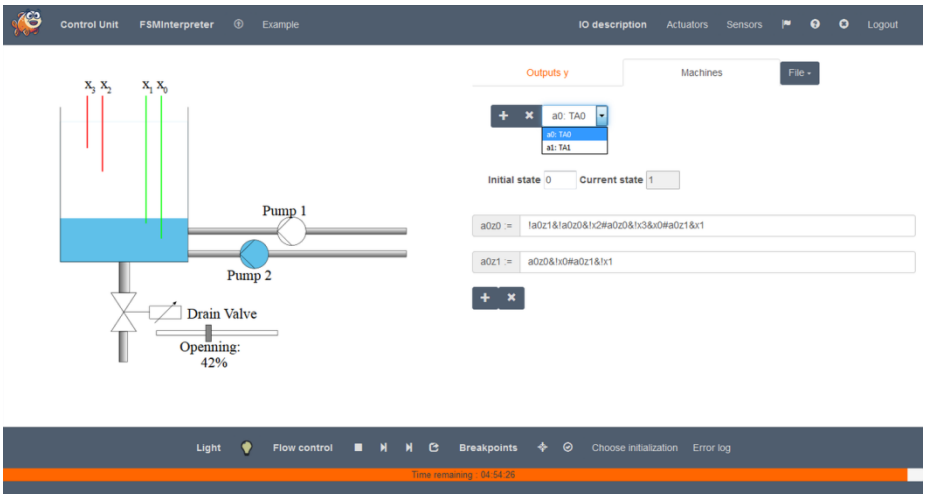

Fig. 16. ECP: Execution of parallel water level control design in virtual mode

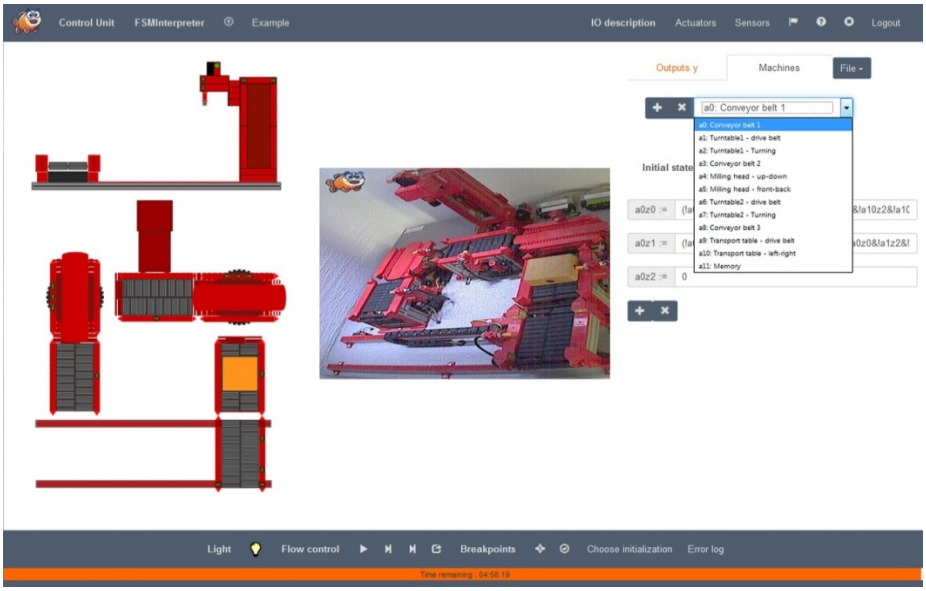

Fig. 17. ECP: Execution of a control task for the "Production Cell" with 12 parallel automata

5 Conclusion

The Integrated Communication Systems Group at the Ilmenau University of Technology is an expert in the field of Internet-supported teaching of digital system design and is well experienced in the area of integrated hard- and software systems. Students have to pass hands-on examinations in a lab to complete the learning outcomes by

own experiences. For all students, hands-on experiences are important to deepen their knowledge about topics they learned during lectures. Therefore they can use the described GIFT system to generate D or JK flip-flop equations to realize their schematics in the hands-on laboratory.

One of the great advantages of the remote laboratory concept to our highly-motivated students is the possibility to test their theoretical knowledge in a real life environment at any time, allowing them to manage their daily schedule more individually and efficient. The GIFT system offers the possibility to export the generated next-state and output equations directly to the Experimental Control Panel of the GOLDi remote lab system, and therefore is an integral part of the GOLDi remote lab architecture as well as the students' learning process. With this possibility to execute their design tasks within our GOLDi infrastructure, we want to offer the students a working environment that is as close as possible to a real world laboratory. Under real laboratory conditions disturbances can appear and lead to failures of the control algorithm that cannot be detected under virtual lab conditions.

6 Acknowledgment

The authors would like to acknowledge the work of Tobias Vietzke, Andrey Yelmanov, Lisa-Marie Schilling, Nicole Ponischil, Lennart Planz, Stephen Ahmad, Bastian Hellweg, Felix Seidel, and David Sukiennik for their work within the GIFT- and GOLDi- framework.

This work was supported in part by the European Commission within the program "Tempus", "ICo-op – Industrial Cooperation and Creative Engineering Education based on Remote Engineering and Virtual Instrumentation", Grant No. 530278-TEMPUS-1-2012-1-DE-TEMPUS-JPHES [10] as well as "DesIRE - Development of Embedded System Courses with implementation of Innovative Virtual approaches for integration of Research, Education and Production in UA, GE, AM", Grant No. 544091-TEMPUS-1-2013-1-BE-TEMPUS-JPCR [11].

We acknowledge support for the Article Processing Charge by the German Research Foundation and the Open Access Publication Fund of the Technische Universität Ilmenau.

7 References

[1] M. Poliakov, H.-D. Wuttke, T. Larionova, K. Henke: "Automated testing physical models in remote laboratories by control event streams," in *Proc. 2016 Int. Conf. Interactive Mobile Communication, Technologies and Learning*, San Diego, CA, USA, October. https://doi.org/10.1109/IMCTL.2016.7753764

[2] K. Henke, T. Vietzke, H.-D. Wuttke, S. Ostendorff: "GOLDi – Grid of Online Lab Devices Ilmenau," *Int. J. Online Eng. (iJOE)*, ISSN: 1861-2121, vol. 12, no. 04, pp.11-13, Vienna, Apr. 2016.

[3] K. Henke, T. Vietzke, H.-D. Wuttke, S. Ostendorff, "GOLDi – Grid of Online Lab Devices Ilmenau," *Demonstration of Online Experimentation, exp.at'15 Int. Conf.* , São Miguel Island, Azores, Portugal, Jun. 2015.
[4] K. Henke, H.-D. Wuttke, T. Vietzke, S. Ostendorff, "Using Interactive Hybrid Online Labs for Rapid Prototyping of Digital Systems," *Int. J. of Online Eng. (iJOE)*, vol. 6, pp. 57-62, Vienna, Oct. 2014.
[5] K. Henke, S. Ostendorff, H.-D. Wuttke, T. Vietzke, C. Lutze, "Fields of Applications for Hybrid Online Labs," *Int. J. of Online Eng. (iJOE)*, vol. 9; pp. 20–30, Vienna, May 2013.
[6] K. Henke, S. Ostendorff, S. Vogel, H.-D. Wuttke, "A Grid Concept for Reliable, Flexible and Robust Remote Engineering Laboratories," *Int. J. Online Eng. (iJOE)*, vol. 8, pp. 42–49, Vienna, Dec. 2012.
[7] K. Henke, T. Vietzke, R. Hutschenreuter, H.- D. Wuttke: "The Remote Lab Cloud "goldi-labs.net"," *13th Int. Conf. on Remote Eng. and Virtual Instrumentation REV 2016*, Madrid, Feb. 2016. https://doi.org/10.1109/REV.2016.7444437
[8] Atmel Corporation: http://www.atmel.com
[9] Altera Corporation: http://www.altera.com
[10] ICo-op project Website: http://www.ICo-op.eu
[11] DesIRE project Website: http://tempus-desire.thomasmore.be
[12] GOLDi-labs cloud Website: http://goldi-labs.net

8 Authors

Karsten Henke (corresponding author) is a scientist and lecturer at the Technische Universität Ilmenau, Department of Computer Science & Automation, Ehrenbergstraße 29, 98693 Ilmenau, Germany. He obtained his PhD at the TU Ilmenau in 1986.

Tobias Fäth is a scientist and lecturer at the Technische Universität Ilmenau, Department of Computer Science & Automation, Ehrenbergstraße 29, 98693 Ilmenau, Germany. He obtained his Master of Science at the TU Ilmenau in 2016.

René Hutschenreuter is a scientist and lecturer at the Technische Universität Ilmenau, Department of Computer Science & Automation, Ehrenbergstraße 29, 98693 Ilmenau, Germany. He obtained his Diploma in computer science at the TU Ilmenau in 2015.

Heinz-Dietrich Wuttke is a senior researcher and lecturer at the Technische Universität Ilmenau, Department of Computer Science & Automation, Ehrenberg- straße 29, 98693 Ilmenau, Germany. He obtained his PhD at the TU Ilmenau in 1983.

This article is a revised version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2017), held in New York, NY, USA, March 2017. Article submitted 01 July 2017. Published as resubmitted by the authors 03 August 2017.