

## C4 Model-Based Design of Embedded Systems

Time: Wednesday, 15.09.2010

Location: Humboldt-Building, Lecture Room 129

Chairmen: A. Zimmermann (DE-Ilmenau)

9:00 a.m.	St. Bosse (DE-Bremen)
<p><b>Hardware Synthesis of Complex System-on-Chip-Designs for Embedded Systems Using a Behavioural Programming and Multi-Process Model</b></p> <p>Embedded Systems used for control, for example in Cyber-Physical-Systems (CPS), perform the monitoring and control of complex physical processes using applications running on dedicated execution platforms in a resource-constrained manner. Traditionally program-controlled multi-processor architectures are used to provide the execution platform. There are two different ways to model and implement System-on-Chip-Designs (SoC) used in those embedded systems: using 1. a structural and/or 2. a behavioural level. The structural level decomposes a SoC into independent submodules - processor cores, memories and peripherals - interacting with each other using centralized or distributed networks and communication protocols. The behavioural level usually describes the behaviour of the full design interacting with the environment, generally a more sophisticated modelling level. In the context of CPS these are mainly reactive systems with dominant and complex control paths. The major contribution to concurrency appears on control path level. A new SoC-design methodology is presented using the behavioural hardware compiler ConPro providing an imperative programming model based on concurrent communicating sequential processes (CSP) with an extensive set of interprocess-communication primitives. The programming language and the compiler-based synthesis process enables the design of constrained power- and resource-aware embedded systems with pure Register-Transfer-Logic efficiently mapped to FPGA and ASIC technologies. Concurrency is modelled explicitly on control- and data path. Additionally, concurrency on data path level can be explored and optimized automatically by different schedulers. The CSP programming model can be synthesized to different other levels, not only used for hardware circuit synthesis: software models (C, ML), intermediate mCode, RTL state level, and finally VHDL. The C and ML output enables a common source for both hardware and software implementation with identical functional behaviour. An extended case study of a communication protocol used in high-density sensor-actuator-networks should demonstrate the design of a SoC for a robot actuator. The communication protocol is suited for high-density intra- and interchip networks.</p>	

9:20 a.m.	A. Osadchuk , W. Fengler (DE-Ilmenau)
<p><b>Survey of Network-on-Chip Research and Modelling Approaches</b></p> <p>The evolution of integrated circuit technology made possible a realization of so-called Multiprocessor System on Chip (MPSoC), where multiple heterogeneous processor cores are integrated on a single chip. As communication infrastructure MPSoC design uses the on-chip networks that have been proposed for System-on-Chip(SoC) integration. The main topic of this paper is to give a research survey over the last years in the field of networks-on-chip (NoC). This paper considers basic guidelines and advantages of NoC research. Some of prototyped NoCs will be presented here. As the subset of SoC, NoC are an integral part of SoC design and like in every design flow the modeling and simulation are important. It is necessary to model a behavior and to realize virtual prototypes of such systems to evaluate the advantages of their technological implementation. However, NoC modeling is still an open research area. Several modeling techniques that are used in NoC design will also be presented and compared in this paper. We make this survey in order to develop an optimal modeling technique for future work.</p>	
9:40 a.m.	B. Däne (DE-Ilmenau)
<p><b>Designing an Application for Field Programmable Gate Arrays – A Case Study</b></p> <p>Field Programmable Gate Arrays (FPGA) are widely used for implementing various functions that previously would have been realized by software. For integration into complex model based design processes it is desirable to derive FPGA designs from high level models such as Matlab/Simulink models. For this purpose manufacturers provide tool support, but some difficulties remain to the user. The paper presents a case study that deals with such a design process while preparing a multiprocessor communication system using multiple FPGA. 2. PROJECT DETAILS Goal of the project that provides the basis for this case study is a decentralized communication system for a multiprocessor system consisting of Digital Signal Processors (DSP). This communication system should provide high throughput while maintaining low software overhead and low pin count. So a fast serial communication protocol is implemented into FPGA (one per processor node), completed by telegram buffers and interfaces to processor bus. Communication topology is a ring. 3. MODELING, CASE STUDY AND RESULTS High level modeling with Matlab/Simulink is used to develop telegram formats, error correction and other aspects of the communication protocol, as well as interfacing of FPGA to DSP. By adding behavioral models of DSP (i.e. bus cycles and test sequences) and error models for the serial lines the protocol is validated and analyzed for effectiveness within the modeling tool. Transformation into actual FPGA design is done by combining tools that are available commercially, and the result is checked against the results that derived from modeling. This process exposes some difficulties that must be addressed when integrating it into more complex design processes. This work partly has been supported by Deutsche Forschungsgemeinschaft (DFG). Matlab and Simulink are trademarks of The MathWorks, Inc.</p>	

10:00 a.m.	A. Pacholik, W. Fengler, T. Simon (DE-Ilmenau)
<b>Case Study: Verification of an Embedded Operating System</b> Verification of complex distributed systems is a challenging task. There are a number of approaches, based on the correctness by design concept, involving code generation. However such approaches are not always feasible. On the other side embedded systems incorporate distributed multitasking systems, are hard to be verified by code analysis alone. In our approach we combine automated code level analysis with high level modeling of selected hardware mechanisms. The fitness of the approach is demonstrated by checking a number of critical properties in a custom real time operating system for a high performance 32-Bit digital signal processor.	
<b>10:20 – 10:40 a.m. Coffee break</b>	
<b>12:00 noon – 1:00 p.m. Lunch</b>	
<b>End of Lecture Session</b>	