Transaction-based Modeling for Large Scale Simulations of Heterogeneous Systems

(Poster Abstract)

Richard Buchmann Mickael Cartron Yannick Bonhomme

CEA, LIST
Laboratoire de Fiabilisation des Systèmes Embarqués
Boîte Courrier 94
Gif-sur-Yvette, F-91191 France
firstname.lastname@cea.fr

ABSTRACT

This paper focuses on the simulation of heterogeneous systems, in terms of hardware and software, that contain thousands of various *Embedded Control Units* (ECUs) which communicate with each other throughout a network. Designers need to simulate ECU interactions to execute and validate the whole system since hardware and software have strong mutual dependencies. We propose a modeling approach based on a *Hardware Abstraction Layer* (HAL) interface, which is the main programming constraint, to achieve fast simulations. Our simulator allows to perform early simulation of heterogeneous ECUs in a few seconds and scales up to thousands ECUs. Experimental results show that the simulation speed and the scalability are very similar to state of the art *Wireless Sensor Network* (WSN) simulators but our approach implies lesser constraints about programming language, and simulation kernel.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids—*Simulation* ; B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids

General Terms

Design, Modeling, Simulation, Performance

1. INTRODUCTION

Many embedded systems are deployed in different domains such as transportation, health, or defense. In those domains, embedded systems mostly execute various distributed applications such as control, monitoring or surveillance. Some of those systems[1] contains hundred of microcontrollers which communicate with each other throughout a network (e.q. CAN, FlexRay, Zygbee, Wavenis...) to execute an application and to provide various services. The complexity of electronic embedded systems is still growing and the dependencies between systems increase.

The system dependencies dramatically increase when executing some distributed applications. For example, in WSN context, localization systems[2], neighborhood discovery protocol, and especially heterogeneous collaborative sensor networks involve a lot of system dependencies. Moreover, hardware and software have strong mutual dependencies such as power-aware algorithms[3] and power management strategies, for example. Such systems are difficult to specify and validate. To overcome dependency problems, designers still need to simulate ECU interactions to execute and validate the whole application. Designers need to map software onto an increasing number of virtual ECUs and consequently they require higher performance from the simulator.

Many approaches and tools focus on simulating a single ECU but the simulations are too slow for distributed applications, like WSN ones, that implicitly run on a large number of ECUs. Simulators like ATEMU[4], Avrora[5], and TOSSIM[6] are intended to simulate a large number (thousands) of communicating ECUs. Each of those simulators targets a particular trade-off between speed, accuracy, heterogeneity and scalability to provide useful simulation results: functional validation, timing and/or consumption estimation for example. Those simulators generally put hard constraints on the framework because they are all-in-one: language, programming style, specific simulation kernel and hardware components are very closely associated. As a consequence, if designers need another trade-off, they must rewrite codes to migrate to the new framework.

This paper focuses on the following aspects of HW/SW co-simulations: early simulation, simulation speed, scalability and especially heterogeneity in terms of both software and hardware. We propose a modeling approach based on a HAL interface which is the main programming constraint.

The next section describes briefly the modeling approach and the chosen abstraction level. We conclude in section 4.

[*] The work reported here was performed as part of the ongoing research program: uSWN FP6-2005 IST-034642 and funded by the European Social Fund (ESF).

SIMUTools 2009 March 2-6, Rome, Italy
Copyright 2009 ICST, ISBN 978-963-9799-45-5.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5769
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5769

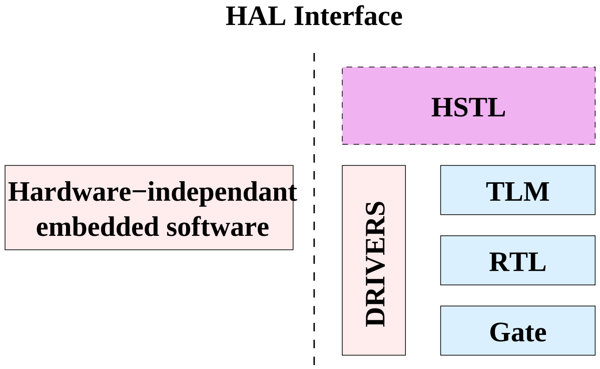

Figure 1: Multi-abstraction level platform

2. MODELING APPROACH

We propose to use a higher abstraction level to speed up the simulation: we model hardware core and the transactions only between hardware and software. Modeling transactions between hardware inside an ECU provides an accuracy that is not necessarily required to execute and validate distributed applications at an early stage. Mechanisms like hardware protocol handshakes are abstracted and pin level is lost.

Our approach is based on a HAL. The HAL interface provides a high level access to hardware components. Its initial function is to abstract the hardware part in order to cleanly separate the application code from the hardware dependent code. This clean separation also allows to describe the hardware at the most relevant level of abstraction for the best simulation trade-off.

Figure 1 depicts a multi-abstraction level platform for the system simulation. The embedded software is hardware-independent and drives the hardware only through a HAL interface. The HAL function may execute drivers code which implies that the CPU will initiate hardware requests to the hardware components (cache, RAM, *Analog to Digital Converter* (ADC), ...). In such a case, the hardware component is described at any abstraction level (e.g. gate level, *Register Transaction Level* (RTL) or *Transaction Level Model* (TLM)[7]).

Our approach is a hybrid abstraction level which abstracts both the driver and the hardware. At *Hardware/Software Transaction Level* (HSTL), simulating does not need accurate hardware descriptions, drivers and the models of CPU and interconnects. HSTL allows designers to model an ECU prior the full HW/SW partitioning because complete hardware mapping and memory mapping are not yet needed to simulate.

3. EXPERIMENTAL RESULTS

One of our experimental bench is a WSN monitoring application and the hardware platform is mono-processor, control oriented, and power constrained. The network is heterogeneous because there are two separate embedded software programs, and two separate hardware platforms. The first type of node is dedicated to monitor the temperature and alert when a temperature fall occurs. The second type is a kind of messenger to get remotely warned when necessary. Simulation speed is similar to TOSSIM.

4. CONCLUSIONS

This work focuses on distributed application simulation on a large number of communicating heterogeneous ECUs. We propose a modeling approach to perform early hardware/ software simulation at high speed. This approach is based on high level hardware/software transactions modeling using a HAL. The proposed framework is intended to help designers to specify and validate complex heterogeneous systems that may include strong mutual dependencies.

HSTL modeling is useful to provide estimations about software performance and hardware consumption over a plain software model in C. We can simulate the whole network as soon as hardware independent software becomes available.

Our experimental results show that simulations of hundred heterogeneous nodes described at HSTL reaches a high speed, similar to the state of the art. In addition, the proposed simulation framework is also flexible because it does not depend on a restrictive programming language unlike TOSSIM, in which system designers must implement both hardware models and embedded software in a single programming language based on events. Moreover, our framework enables heterogeneity in terms of both software and hardware, and also supports simulating at multi-abstraction level: Designers may choose independently the best abstraction level for each hardware component to obtain the target trade-off between speed, accuracy and scalability.

5. REFERENCES

[1] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi. MobEyes: Smart Mobs for Urban Monitoring with a Vehicular Sensor Network. *IEEE Wireless Communications*, 13(5):51–57, 2006.

[2] S. Sundresh, W. Kim, and G. Agha. SENS: a sensor, environment and network simulator. *Simulation Symposium, 2004. Proceedings. 37th Annual*, pages 221–228, 2004.

[3] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 181–190, New York, NY, USA, 1998. ACM.

[4] J. Polley, D. Blazakis, J. McGee, D. Rusk, and JS Baras. ATEMU: a fine-grained sensor network simulator. *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 145–152, 2004.

[5] B.L. Titzer, D.K. Lee, and J. Palsberg. Avrora: scalable sensor network simulation with precise timing. *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005.

[6] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: accurate and scalable simulation of entire TinyOS applications. *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137, 2003.

[7] A. Rose, S. Swan, J. Pierce, and J.M. Fernandez. Transaction Level Modeling in SystemC. *Open SystemC Initiative*, 2005.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5769
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5769