ARTES Project Proposal

Design of Heterogeneous Multiprocessor Systems for Real-Time Applications

Petru Eles and Zebo Peng
Embedded Systems Laboratory (ESLAB)
Dept. for Computer and Information Science
Linköping University

Summary

This project is planned as part of a large ongoing effort carried out at ESLAB, Linköping University, aimed to develop methodologies and tools for the design of real-time applications implemented as multiprocessor systems.

The main focus of the proposed research will be on design methodologies and tools for heterogeneous systems in telecom applications, consisting of several programmable processors and dedicated hardware components. One of the main goals is to support the designer from the early phases of the design, in which the application is specified at a high-level of abstraction, and to support the automatic and/or interactive exploration of a large design space, which leads to high performance and cost efficient solutions.

In particular, methods will be investigated in order to support the selection of a system architecture which optimally suits a given application, so that costs are minimized and design constraints are fulfilled. This implies the selection of programmable processors, dedicated hardware components, memories, busses and interconnection topology. The proposed strategy, which has also to facilitate the reuse of earlier designs as part of the new architecture, will be based on a hierarchical architecture model and component libraries. An important aspect of this research will be the development of fast and accurate estimation tools for performance and cost, which have to guide the design exploration process towards an efficient implementation. Another aspect of the research concerns the investigation of scheduling alternatives for embedded hardware/software systems.

The basestation in a digital mobile-telephone network has been defined as a demonstrator application to be used for this project.

We would like to apply for funding to support one Ph.D student for a period of four (2 + 2) years. The research will be done in cooperation with the industry and will use industrial examples to validate the proposed concepts.

1. Introduction

It is becoming increasingly difficult to design embedded computer systems for industrial applications due to their complexity and design requirement. Many of such applications have great demand on the performance of their underlying systems which can only be delivered by a multiprocessor architecture. Studies of the general features of such architectures and the selection and design of an appropriate architecture for a given application have therefore become an important research area [2, 3, 9, 11].

The main objective of the proposed project is to develop a methodology and tools to support the design of embedded system architectures based on multiprocessor models. In particular, we will focus on issues related to real-time applications where a set of timing constraints is imposed on the implementation. We will also concentrate on design steps at the system level, such as architecture selection, performance estimation, scheduling and allocation.

Our main approach is to formulate the design tasks as optimization problems and to develop efficient heuristics to solve them. For embedded real-time applications, a cost function to drive the optimization process can often be defined since the designed system is targeted for a given application. Our intention is also to customize general optimization techniques to suit the characteristics of applications in an area in order to make them more efficient. We have decided to target the telecom application area which combines control aspects with heavy data manipulation. Our research will therefore be carried out in close cooperation with the Swedish telecom companies.

2. Objectives

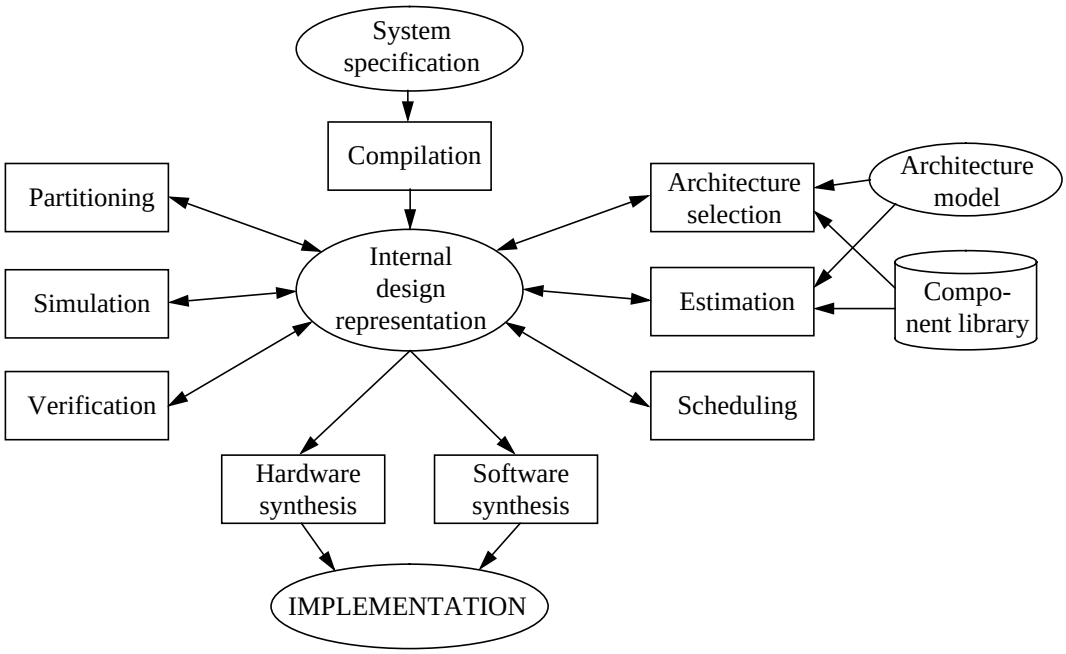
The proposed research is planned as part of an ongoing effort aimed to develop methodologies and tools for the design of real-time applications implemented as multiprocessor systems. In order to provide a cost efficient solution which at the same time satisfies the imposed performance requirements, such multiprocessor systems very often have to be heterogeneous in their structure. This means that they consist of both programmable processors and dedicated hardware components. Further heterogeneity can be identified inside each of these two domains. For example, programmable processors can be general purpose microprocessors, microcontrollers or digital signal processors (DSPs). The dedicated hardware components on the other side can be implemented as application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs). Our final goal is to develop a design environment which supports the specification, verification, analysis, and synthesis of heterogeneous multiprocessor systems for real-time applications. This ambitious goal covers research topics in various areas which have been the object of several of our completed projects [1, 5, 6, 8, 12], while other projects are currently carried out. Thus, our proposed project will benefit from several previously developed algorithms and tools and will be carried out in strong synergy with other complementary research (see section 7).

The following figure shows the structure of the proposed design environment [7]. One of the important features of this design environment is that it starts from specifications at a high level of abstraction and supports the designer from the early phases of the design. The input specification only describes the overall functionality of the designed system and includes the imposed constraints (timing, power consumption, cost, etc). Starting from such a system specification, the design environment supports the automatic or interactive exploration of a large design space which leads to high performance and cost effective solutions. At the same time, early detection of design errors and the use of synthesis tools which take into consideration timing constraints, drastically reduces the number of functional and timing errors detected in later design phases and, thus, avoids expensive revisions of the design.

In the following we highlight those topics which are relevant for this research proposal.

Internal abstract representation

The internal design representation (IDR) constitutes the back-bone of the system level



design tasks. As it can be observed from the figure above, synthesis tools working at the system level perform their analysis and synthesis tasks on this abstract representation. The input system specification[1] will be compiled into this representation which then is progressively refined and annotated as more and more implementation details are added by the system level design tools.

The IDR will be a hierarchical representation at two levels:

1. The system level: at this level the system will be represented as a graph capturing the tasks and their interaction both in terms of dataflow and in those of control. The representation should be adequate for simulation as well as for analysis and synthesis purposes during architecture selection, partitioning and scheduling.
2. The task level: the representation at task level has to capture information at a lower level of granularity. Analysis activities at the level of individual processes (like execution time estimation) will be performed on this representation.

Architecture Selection

Architecture selection is aimed to define the number and kind of programmable processors, dedicated hardware components, shared and private memories, and busses, as well as the interconnection topology. One important aspect of this problem is to develop a methodology which facilitates the reuse of earlier designs as part of the new architecture.

Architecture selection is based on models situated at two hierarchical levels:

- an abstract architecture model which defines the class of architectures taken into consideration; it is very important that such a model captures the essential features of the architectures which are relevant to a certain application area; therefore we plan to closely cooperate with the industrial partner concerning this modeling aspect;
- a component library containing processor, bus, and memory models, and also models corresponding to earlier reusable designs.

We believe that a transformational design approach in which partitioning and architecture selection steps are performed iteratively and are guided automatically by an optimization heuristic strategy and/or by user interaction is a way to solve this problem. It is very important to provide an environment in which the experience and skills of the individual designer are

1. Currently our design environment accepts VHDL as a specification language [4, 7]. We plan to finally implement front-end tools to accept input specifications in various languages.

combined with the power of design automation tools in order to identify the proper architecture for a given application. Thus, several elements of the architecture could be fixed by the designer, narrowing the design space which remains to be explored with support of the design environment. An alternative strategy is to let the design tool to identify, based on certain heuristics, areas of the design space which should be further explored and analyzed by the designer.

Performance and cost estimation

Estimation tools are among the key components of a system design environment. Any decision taken during the design process, regardless if this decision is taken automatically or as result of user interaction, is based on an estimation of certain features of the current design alternative. In this project we plan to investigate mainly performance and cost estimation aspects:

1. Process execution time estimation is based on the processor model and on the IDR at process level. It should also take into consideration aspects concerning cache memory and pipeline hazards and their influence on the predictability of system performance.
2. Estimation of the timing behaviour of the system is strongly connected to process scheduling. A highly accurate estimation will be based on an analysis of process scheduling using estimated process execution times, as well as estimations of communication delays. However, in order to be used inside an iterative optimization process, we also have to develop faster algorithms based on an incremental estimation. The basic idea is to estimate the change in timing behavior resulted after a certain partitioning or allocation step. Thus, a hierarchy of estimation algorithms can be developed, from fast and less accurate to slower and highly accurate, together with a strategy to apply the most appropriate one at different stages of the design space exploration.
3. Cost estimation is related to design parameters such as area, number of pins, memory requirements, busses, number and cost of programmable processors, etc. Their influence on the final cost depends very much on characteristics of the application area and on market specific aspects. We plan to closely cooperate with our industrial partner in this respect.

As part of the estimation problem we have also to consider that at certain stages of the design process the specification is still incomplete. Under such circumstances estimation has to be performed in close interaction with the designer and by exploration of data referring to previous similar designs.

Process scheduling

Process scheduling is decisive from the point of view of the final system performance. For scheduling we have to use information about the mapping of processes, as well as estimated (worst case) execution times of processes and communication delays. Given the distributed nature of the system, sharing of busses as well as memory conflicts have to be considered. Heterogeneousness introduces another dimension of complexity as we have to consider the interaction of software and hardware components. We plan to implement several alternative strategies from which the designer can choose based on the specific features of the application. Such are fixed priority scheduling as well as static non-preemptive scheduling. The main goal is to provide high predictability of the systems behaviour, satisfaction of hard real-time constraints, and low system overhead.

3. Expected Results

The following main results will be generated as an outcome of this research effort:

- An IDR which captures a system level view of the application and will be refined to include increasing levels of detail as the system design process evolves.
- A methodology and tools for modeling of multiprocessor architectures and processors used for embedded real-time systems; tool support for architecture selection, based on user interaction and on automatically guided design space exploration.
- Estimation techniques and tools for performance and cost analysis at process and system

level. Estimation will provide the necessary feedback to evaluate a certain design alternative and to decide on further design steps.
- Process scheduling strategies and tools providing predictability, satisfaction of real-time constraints and low system overhead.

4. Project Plan

The main research will be carried out by a PhD student with active involvement of some industrial partner which participates in the formulation of the problems, selection of industrial examples, and evaluation of results. The student is planned to spend 3 months at Ericsson Radio in a mobility program in order to become familiar with the particularities of the application area and the industrial development process.

The following problems are planned to be investigated during the first two years:
- Modeling techniques for the component library have to be developed. The abstraction level has to be suitable for efficient performance estimation and should support reusability.
- Modeling techniques for heterogeneous multiprocessor architectures based on the particularities of the application area.
- Estimation techniques will be developed at the process level and at system level. This task will be carried out in cooperation with the work in our hardware/software co-design project (see section 7).

The work after the first two years will be documented as a Lic. thesis.

For the additional two years we plan continuation of the work on scheduling algorithms and the development of automatic and interactive tools for architecture selection. Automatic design space exploration for architecture selection and process mapping can be based on the guidance of optimization heuristics which will also be developed.

5. Application:
The Basestation in a Digital Mobile-telephone Network

In co-operation with our industrial partners we will target our research towards some specific needs of the telecom application area. We have decided to define *the basestation in a digital mobile-telephone network* as a demonstrator application which will be used to validate some of the developed methodologies and tools. As results from the following short description, the basestation functionality should be efficiently implemented as a heterogeneous hardware/software architecture after exploration of several possible alternative solutions, along the concepts outlined in the previous sections.

One of the nodes in a digital mobile-telephone network is the basestation. Its task is to provide the mobile-telephone (further called simply *mobile*) with an interface to the network. The mobile communicates with the basestation via radio. The basestation is, in its turn, connected to a controller in the network through a suitable medium, e.g., PCM-line, radio, or optic fiber. The communication between the basestation and the mobile is controlled by the controller to a large extent but in some standards the basestation has the authority to make some of the decisions, e.g., to disconnect a call.

The basestation receives three kinds of information from the controller: 1) control information that tells it how to communicate with the mobile; 2) signalling information that should be relayed to the mobile; 3) voice data that should be relayed to the mobile. From the mobile it receives: 1) measurement data that is used to control the transmission; 2) voice data that is to be relayed to the controller; 3) signalling information that should be relayed to the controller node.

A simple way of describing the structure of the basestation is to say that it consists of

two pipes, one for transmission and one for reception. Each pipe is made up of a chain of signal-processing units controlled by control units. Each pipe has an analogue part and a digital part, with the former being closest to the antenna. On its way through the transmission pipe, voice data has to be coded, encrypted, modulated and amplified. Somewhere along the line it is converted to an analogue signal. The receiving pipe is very much the mirror of the transmission pipe. Signalling data for the mobile and measurement data from it are treated in a similar fashion.

The speed at which the different operations have to be carried out varies considerably since a signal is transformed between different frequency bands as it passes through the basestation. A further complication is that several mobiles may be communicating with the same basestation at the same time. In the future, things will be even more complicated since basestations will be expected to handle multiple communication standards.

To implement a basestation several media are used: analogue hardware, digital hardware, microcontroller software, and DSP software. The choice of media blend in any particular system is a trade-off between production cost, speed requirements imposed by the standards, signal-processing characteristics, flexibility, and so on. In general, there is a trend to move more and more of the functionality into the digital domain, and more and more of the digital domain into software.

A typical basestation of the future will contain several co-operating signal processors, control processors, dedicated hardware (analogue and digital), multiple buses, and last but not least, considerable amounts of memory.

Some of the more notable bottle-necks in basestation design, which make it a very interesting example in this context, are: 1) bus speed contra the size and power consumption of the buses; 2) processor speed; 3) the speed of AD/DA converters; 4) the non-linearity of the analogue parts, which inflict higher demands on the digital parts; 5) production cost; 6) size; 7) flexibility and architecture-independence needed to meet the ever-increasing speed of changes in technology and standards.

In summary, a basestation is a typical heterogeneous system. This is true both for the choice of implementation media, and for the operations that it performs. This puts demands on all steps of the construction of a basestation: specification, design, evaluation, implementation, production, and testing.

6. Preliminary Budget

The preliminary yearly budget is calculated as follows:

Ph.D. Student, 80%	300.000 SEK
Supervisors (Eles, Peng), 10%	70.000 SEK
Computer and equipment	50.000 SEK
Travel	30.000 SEK
Administration and office	50.000 SEK
Total budget per year:	500.000 SEK

The project will last for 2 + 2 years.

7. The Research Group and Related Activities

The Embedded Systems Laboratory (ESLAB) conducts research on the design and test of embedded systems, especially those consisting of tightly coupled hardware and software components. Special emphasis is placed upon the development of methods and tools for specification, modeling, synthesis, simulation, design for test, test synthesis, and hardware/software

co-design. We are also concerned with the exploitation of systematic design methods and design automation techniques for industrial applications.

ESLAB is currently involved in several projects covering a wide spectrum of research areas. We summarize some of these projects, which are closely related to the proposed research: ***Hardware/software co-design of embedded systems***. This is an ARTES project which is being carried in cooperation with Saab Dynamics AB, Saab AB and Volvo Technological Development. It aims at the development of methods and tools for the architecture selection problem in hardware/software co-design for control dominated applications. This entails formulating a hardware/software architecture model, developing a mapping strategy to map a real-time system specification into such an architecture, and designing methods and tools for the analysis of a given design based on the given architecture. We study also the problem of how to use the analysis results to guide the modification of the given architecture in such a way that the final architecture corresponds to the optimal design of the system specification.

Design of control and communication dominated system. ESLAB has been involved in a joint research project, together with the Electronic System Design Lab at KTH, Synthesia AB, Ericsson Telecom, and Saab Dynamics, which is sponsored by NUTEK. The project aims at the development of a design environment for control and communication dominated embedded systems consisting of hardware and software components. Based on an evaluation model, a specified system is partitioned into hardware and software domains and implementation details are added. We are also developing techniques to support re-partitioning and to handle changes in design requirements or of technology.

8. Industrial Cooperation

The main industrial partner of this project is Ericsson Radio Systems AB. It has participated in the formulation of the research direction and the demonstrator application, in the identification of the main research issues of the project and will be involved partially in the execution of the research as well as its case studies and evaluations.

Contact information for the industrial partner is as follows:

Erik Stoy
X/R, Ericsson Radio Systems, 164 80 Stockholm
Erik.Stoy@era.ericsson.se, Tel: 08-585 313 42, Fax: 08-404 53 32

List of Appendices to the project "Design of Heterogeneous Multiprocessor Systems for Real-Time Applications"

1. References.

2. CV of Dr. Petru Eles.

3. Short resume of Prof. Zebo Peng.

4. Supporting letter from Ericsson Radio Systems AB.

References

1. Axelsson J., "Analysis and Synthesis of Heterogeneous Real-Time Systems", Ph.D Dissertation, Linköping University, Sweden, 1997.
2. Camposano R. and Wilberg J., "Embedded System Design", *Design Automation for Embedded Systems*, vol. 1, no. 1-2, January 1996, pp. 5-50.
3. Edwards S, Lavagno L., Lee E. A., and Sangiovanni-Vincentelli A., "Design of Embedded Systems: Formal Models, Validation, and Synthesis", *Proceedings of the IEEE*, vol. 85, no. 3, March 1997, pp. 366-390.
4. Eles P., Kuchcinski K. and Peng Z., "Synthesis of Systems Specified as Interacting VHDL Processes", *Integration, the VLSI Journal*, 21 (1996) 113-138.
5. Eles P., Peng Z., Kuchcinski K., and Doboli A., "Hardware/Software Partitioning with Iterative Improvement Heuristics", Proceedings of the 9th International Symposium on System Synthesis, San Diego, 1996, IEEE Computer Society Press, pp. 71-76.
6. Eles P., Peng Z., Kuchcinski K., and Doboli A., "System Level Hardware/Software Partitioning Based on Simulated Annealing and Tabu-Search", *Design Automation for Embedded Systems*, 2, 5-32 (1997).
7. Eles P., Kuchcinski K. and Peng Z., *System Synthesis with VHDL*, Kluwer Academic Publishers, Boston, 1998.
8. Eles P., Kuchcinski K., Peng Z., Pop P., Doboli A., "Scheduling of Conditional Process Graphs for the Synthesis of Embedded Systems", Proceedings of Design, Automation and Test in Europe (DATE), Paris, 1998, IEEE Computer Society Press, pp. 132-138.
9. P. G. Paulin et al., "Trends in Embedded System Technology: an Industrial Perspective", in G. De Micheli and M. G. Sami, eds: *Hardware/Software Co-Design*, NATO ASI 1995, Kluwer Academic Publisher, 1996.
10. Z. Peng and K. Kuchcinski, "Automated Transformation of Algorithms into Register-Transfer Level Implementation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 2, Feb. 1994, pp. 150-166.
11. W. Wolf, "Hardware-Software Co-Design of Embedded Systems", *Proceedings of the IEEE*, vol. 82, no. 7, July 1994, pp. 967-989.
12. Stoy E and Peng Z., "Inter-Domain Movement of Functionality as a Repartitioning Strategy for Hardware/Software Co-Design", *Journal of Systems Architecture*, vol. 43, 97-98, 1997.