



Universiteit  
Leiden  
The Netherlands

## Improved hard real-time scheduling and transformations for embedded Streaming Applications

Spasic, J.

### Citation

Spasic, J. (2017, November 14). *Improved hard real-time scheduling and transformations for embedded Streaming Applications*. Retrieved from <https://hdl.handle.net/1887/59459>

Version: Not Applicable (or Unknown)

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/59459>

**Note:** To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The following handle holds various files of this Leiden University dissertation:

<http://hdl.handle.net/1887/59459>

**Author:** Spasic, J.

**Title:** Improved hard real-time scheduling and transformations for embedded Streaming Applications

**Issue Date:** 2017-11-14

## Chapter 7

# Summary and Conclusions

The continuous increase in user demands and very fast technology improvement have led to more and more complex embedded streaming systems. Nowadays, many embedded systems are based on Multi-Processor System-on-Chip (MPSoC) platforms. In modern MPSoCs, it is desirable to execute multiple applications, a mixture of streaming applications and control (hard) real-time applications, simultaneously in order to efficiently utilize the resources in an MPSoC. To deliver high-quality output of multiple running applications, together with the ability to dynamically start/stop applications without affecting other already running applications, streaming applications have tight timing requirements that often make it necessary to treat them as hard real-time applications. Moreover, given that the embedded systems are very often battery-powered, a very important requirement in the design of embedded streaming MPSoCs is the energy-efficiency. Designing such an embedded system imposes several challenges: a streaming application should be represented in a way that reveals the parallelism of the application, and it should be mapped and scheduled on a platform such that the timing requirements are satisfied and the energy consumption minimized. To exploit the parallel nature of MPSoC platforms, application behavior is usually specified using a certain parallel Model of Computation (MoC), in which the application is represented as parallel executing and communicating tasks. Finding an efficient tasks-to-platform mapping, that is, *spatial scheduling*, and execution order of tasks in time, that is, *time scheduling*, are the key issues for optimizing the energy consumption and performance of these systems.

In Chapter 3, we have proposed a *conversion approach* that converts tasks in a MoC to real-time tasks to solve the problem of providing timing guarantees for embedded streaming systems. In particular, we have devised an

improved hard real-time scheduling approach to schedule streaming applications modeled as acyclic CSDF graphs on an MPSoC platform. Our proposed approach converts each actor in a CSDF graph to a set of real-time periodic tasks. The conversion enables application of many hard real-time scheduling algorithms that offer fast calculation of the required number of processors for scheduling the tasks. In addition, in Chapter 3, we have proposed a method to reduce the graph latency when the converted tasks are scheduled as real-time periodic tasks. Our proposed scheduling approach gives tighter guarantee on the throughput and better processor utilization with acceptable increase in terms of communication memory requirements when compared with related scheduling approaches. Although it has been shown in [TA10] that the majority of streaming applications, around 90%, can be represented as acyclic SDF graphs, extending the scheduling framework presented in Chapter 3 to support streaming applications modeled as cyclic (C)SDF graphs deserves further investigation.

To exploit efficiently the available parallelism in an MPSoC platform to guarantee performance, energy, and timing constraints, the right amount of parallelism available in a streaming application should be exposed. Given that an initial application specification is often not the most suitable one for the given MPSoC platform, in Chapter 4, we have proposed an *unfolding transformation approach* to transform an initial application specification into an alternative application specification which closely matches the given MPSoC platform. Our proposed approach transforms an initial SDF graph to an alternative CSDF graph which exploits the proper amount of parallelism in the given homogeneous MPSoC to maximize the system performance and provide timing guarantees. The tasks in the transformed graph are scheduled according to our scheduling approach presented in Chapter 3. The experiments on a set of real-life applications showed that our proposed approach delivers, in a matter of minutes, solutions with smaller code size, smaller buffer sizes, and shorter application latency while meeting the same performance and timing requirements as the related approaches.

In Chapter 5, we have investigated and proposed an approach which utilizes our unfolding transformation presented in Chapter 4 and our scheduling framework given in Chapter 3 to better map the given streaming application to the given MPSoC platform in terms of energy consumption under throughput constraints. Our *energy-minimization mapping approach* applies the unfolding transformation to an initial SDF graph to the extent that when combined with Voltage-Frequency Scaling (VFS) techniques on cluster-heterogeneous MPSoCs leads to an energy-efficient design which also meets throughput con-

strains. In particular, our proposed approach is a polynomial-time approach which first determines a processor type for each task in an SDF graph such that the throughput constraint is met, and second, determines a replication factor for each task in an SDF graph to achieve load-balancing on processors of the same type, which enables processors to run at a lower frequency, thereby consuming less energy. The experimental evaluation performed on a set of real-life streaming applications showed that our approach reduces energy consumption by 66% on average among all the experiments while meeting the same throughput requirement when compared to related energy minimization mapping approaches.

Finally, in Chapter 6, we have proposed a novel *very accurate energy model* for streaming applications modeled as PPN graphs and mapped onto tile-based MPSoC platforms with distributed memory. The energy model is based on the well-defined properties of the PPN application model. To guarantee the accuracy of the energy model, values of important model parameters were obtained by real measurements. The proposed energy model is applicable to different types of processors and communication infrastructures within an MPSoC platform. The energy model was evaluated on FPGA-based MPSoC platforms against real measurements of the energy consumption from the FPGA. The obtained energy consumption estimates are highly accurate with an average error of 4% and a standard deviation of 3%. The average model evaluation time per design point takes 2.5 minutes for the considered cases, which is very good given the high accuracy of the model. The majority of the evaluation time is spent in SystemC cycle accurate simulations, which are used for obtaining the *time parameters* of the energy model. Although they lead to highly accurate estimates of the energy consumption, these simulations might lead to significantly high overall design space exploration time. Thus, to maintain the high estimation accuracy while providing time-efficient evaluation of the energy consumption, a future work could combine our energy model based on SystemC simulations with a model based on less accurate but faster techniques to estimate the time parameters. The two energy models should be properly interleaved in the design space exploration process to achieve an adequate trade-off between the model accuracy and evaluation time. In that case, our highly accurate energy model would be used to estimate the energy consumption of a set of design points preselected by using the less energy accurate model in the process of design space exploration.

