Chapter 8

Summary and Outlook

MODERN streaming applications exhibit increasingly adaptive behavior and required more complex computation. At the same time, high performance requirements and tight hard real-time guarantees must be satisfied. Driven by the advance of the semiconductor technology, MPSoC platforms will continue to have more computational capabilities to meet these demands. This situation asks for a correspondingly advanced design methodology to cope with the design complexity. Adoption of a model-based ESL design methodology seems to be an inevitable solution to increase the design productivity and handle the complexity. In this thesis, we have proposed several novel techniques to enhance a model-based ESL design methodology. In particular, we have adopted a specific instance of such design methodology, namely the Daedalus^{RT} design flow, for designing high performance, hard real-time, and adaptive streaming systems. Below, we provide a summary of this thesis.

In Chapter 3, we have presented an algorithm to derive a CSDF graph as the analysis model from an input-output equivalent PPN. Automated derivation of the CSDF MoC allows for the subsequent step of HRT analysis in Daedalus^{RT}. Therefore, it is a key enabler of the fully automated Daedalus^{RT} design flow. In addition, our approach can be considered as an enhancement to the PNgen [125] compiler, which derives an equivalent PPN from a SANLP. Now, all design flows that accept the CSDF MoC as application specification will benefit from our approach because it relieves designers from manual specification of the CSDF MoC, which in some case may not be trivial to do it manually.

In Chapter 4, we have shown that the mapping of streaming applications considering a single initial application specification cannot fully utilize the processing power of MPSoC platforms. Using the PPN MoC as the application specification, we have presented an analytical framework to determine the maximum DLP, i.e.,

the maximum number of communication-free partitions. Subsequently, we have proposed an approach to transform an initial PPN to a set of communication-free partitions, if it exists. The experimental results on FPGA-based MPSoCs and desktop multi-core platforms showed that our approach leads to significantly better performance than the approaches, in which alternative application specifications are not taken into account.

In Chapter 5, we have addressed the problem of exploiting just-enough parallelism when mapping a streaming application modeled using the SDF MoC in hard real-time systems. Exploiting just-enough parallelism is achieved by simultaneously unfolding and allocating the SDF actors onto an MPSoC platform, while considering the number of available PEs and hard real-time scheduling of actors on the PEs. We showed that the solution space to our problem is bounded and subsequently derived its upper bound. We devised an efficient algorithm to solve the problem and evaluated the algorithm on a set of real-life applications. The experiments showed that our algorithm results in a system specification with large performance gain. We also compared our algorithm with one of the state-of-the-art meta-heuristics, i.e., NSGA-II genetic algorithm, and showed that our algorithm is on average 100 times faster than the GA, while achieving the same quality of the solution.

In Chapter 6, we have introduced the Parameterized Polyhedral Process Network (P^3N) MoC that is able to capture adaptive/dynamic application behavior. Such behavior is usually expressed by parameters which values are updated at run-time. We have proposed a design-time approach to enable consistent execution of the P^3N MoC at run-time. The P^3N MoC is used as the implementation model for adaptive streaming applications in the Daedalus^{RT} design flow. Therefore, we have evaluated the possible run-time overhead caused by the parameterization of the P^3N model by designing and executing MPSoCs on an FPGA-based platform. The obtained results have shown that the parameterization we proposed is efficient in terms of the execution overhead introduced by the implementation of the process networks.

In Chapter 7, we have introduced the Mode-Aware Data Flow (MADF) MoC for adaptive streaming applications and its operational semantics. An adaptive streaming application is characterized by individual scenarios and scenario transitions. As an important part of the operational semantics, we proposed a novel protocol for scenario transitions. The main advantage of this transition protocol is that it does not introduce any timing interference between scenarios upon transitions. Based on the transition protocol, we have extended the initial HRT scheduling in Daedalus^{RT} for the MADF MoC. Furthermore, we have presented a HRT analysis on best-case and worst-case transition delays. Finally, we have conducted a case study using a real-life adaptive streaming application. The results have shown reasonable increase

of transition delay due to our proposed transition protocol.

Although the techniques proposed in this thesis have significantly enhanced our model-based ESL design methodology, some interesting open research problems still exist that are highly related to the work presented in this thesis. Below, we outline some important ones.

Trade-off Exploration between Communication and Load-balancing

The communication-free partitioning developed in Chapter 4 only considers one special alternative application specification, i.e., no communication between PEs when all communication-free partitions are mapped on them. However, this comes at a cost: the workload on PEs may not be perfectly balanced any more. It is thus worthwhile considering other alternative application specifications as well, in which certain degree of communication between partitions is allowed while workload can be better distributed among partitions. The degree of communication between partitions depends on the computation and communication capabilities of target architectures. We can only have an optimum mapping when all these factors are taken into account in a design space. Clearly, even a larger design space will ask for more efficient DSE algorithms.

Equivalence between the Analysis MoC and the Implementation MoC for Adaptive Applications

It should not be too difficult to derive the P^3N MoC developed in Chapter 6 from a sequential specification based on the initial work in [92]. It is, however, important to show that, for any P^3N, we can find an input-output equivalent MADF graph. Only in this case, we can achieve a complete design flow for adaptive streaming applications as the Daedalus^{RT} design flow for the static streaming applications.

Optimization of Mode Transitions Considering Mapping

The HRT scheduling framework developed in Chapter 7 for the MADF MoC does not yet consider the effect of resource allocation and actor mapping. This immediately brings up an important problem: given a MADF graph and a fixed platform, e.g., m PEs, find a mapping of actors onto m PEs, such that transition delays are minimized while HRT constraints are met.