Task Concurrency Management Methodology Summary

Chun Wong (†,**) Paul Marchal (†,**) Peng Yang (†,**) Aggeliki Prayati (†,‡)
Nathalie Cossement (*) Francky Catthoor (†,***) Rudy Lauwereins (*)
Diederik Verkest (†) Hugo De Man (†,***)

IMEC, Kapeldreef 75, Leuven, Belgium (†), email: catthoor@imec
Department of Electrical and Computer Engineering, University of Patras, Greece (‡)
Katholieke Universiteit Leuven-ESAT, Leuven, Belgium (*)
also Ph.D. student of K.U.Leuven-ESAT (**), also professor of K.U.Leuven-ESAT (***)

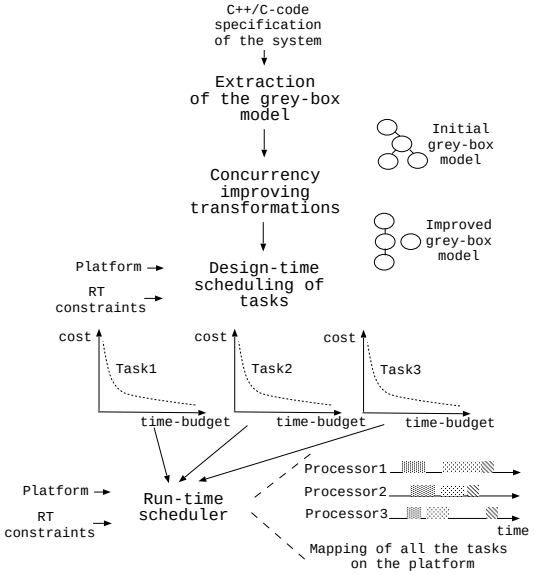

Figure 1. Framework of the task concurrency management methodology

This paper summarizes a new methodology for the design of concurrent dynamic real-time embedded systems. The framework of our methodology is depicted in Fig. 1. An embedded system can be specified at a *grey-box* abstraction level in a combined MTG-CDFG model [6]. The *grey-box* model is different from both the detailed *white-box* model [1] where all the operations are considered during the mapping and where too much information is present to allow a system wide exploration, and the *black-box* model [2, 3] where insufficient information is available to accurately steer even the most crucial cost trade-offs. In contrast, the *grey-box* specification is functional in representing the concepts of concurrency, timing constraints and interaction at either an abstract or a more detailed level, depending on what is required to perform a thorough exploration of the decisions afterwards. We believe that task concurrency management can be implemented in four major steps [4]. Firstly, the *grey-box* model is built, including the necessary concurrency extraction. Then transformations are applied on the specified MTG-CDFG to increase the opportunities for concurrency exploration and cost minimization [5]. Then static scheduling will be applied on the design-time analyzable parts of the *grey-box* model, including processor assignment in the multiple processor context. Finally, a dynamic scheduler will schedule the dynamic and coarse-grain constructs at run time on the given platform while making trade-offs based on Pareto curves.

The main driver for our work is the MPEG-4 IM1 player. Experiment results confirm the validity of our assumptions and the usefulness of our approach [4, 5].

References

[1] F. Balarin *et al*, *Hardware-Software Co-Design of Embedded Systems: the POLIS Approach*. Kluwer Academic Publishers, 1997.

[2] I. Hong, M. Potkonjak, and M. B. Srivastava, "On-Line Scheduling of Hard Real-Time Tasks on Variable Voltage Processor", *International Conference on Computer-Aided Design*, San Jose, 1998, pp. 653-656.

[3] B.Dave and N.Jha, "CASPER: concurrent hardware-software co-synthesis of hard real-time aperiodic and periodic specifications of embedded system architectures", *Proc. 1st Design and Test in Europe*, Paris, Feb. 1998, pp. 118-124.

[4] A. Prayati, C. Wong, P. Marchal *et al*, "Task Concurrency Management Experiment for Power-efficient Speed-up of Embedded MPEG4 IM1 Player", International Workshops on Parallel Processing 2000, Toronto, Aug. 2000, pp. 453-460.

[5] P. Marchal, C. Wong, A. Prayati *et al*, "Impact of task-level concurrency transformations on the MPEG4 IM1 player for weakly parallel processor platforms", International Conference on Parallel Architectures and Compilation Techniques 2000, Philadelphia, Oct. 2000.

[6] N. Cossement, R. Lauwereins, and F. Catthoor, "DF*: An extension of synchronous dataflow with data dependency and non-determinism", *Forum on Design Languages (FDL)*, Tuebingen, Sep. 2000.