報告番号	※甲　　第　　号

主　論　文　の　要　旨

論文題目　Efficient System-Level Design Space Exploration for Large-Scale Embedded Systems
(大規模組込みシステムにおける効率的なシステムレベル設計空間探索)

氏　名　安藤友樹

論　文　内　容　の　要　旨

This dissertation presents system-level design space exploration method for large-scale embedded systems.

Embedded systems have been increasing their complexities, and they are getting more and more functionalities. Since the functionalities of embedded systems have increased, embedded systems need more computational power and communications capacity. Thus, several processing elements (PEs) such as dedicated hardware and multi-core processors have been used in embedded systems.

Embedded systems are generally required to satisfy strict requirements of the system performances such as execution time, chip area, and power consumption since they are used in limited environments. During the design of embedded systems using several PEs, designers must determine a mapping which indicates the allocation of functions to these PEs. Since the system performances depend on the mapping, it is important for designers to find appropriate mappings efficiently from the large design space.

In order to design complex embedded systems, system-level design has been proposed. The key points of system-level design are to design a system at a high level of abstraction and to explore the design space. Designers first describe the system in particular model at a high level of abstraction. Then, the model is converted to a simulation description and a target implementation in order to evaluate the system performances. If the evaluation results of the system performances do not satisfy the requirements, the designers modify the model, and they again evaluate the system performances. Designers iterate modification of the model and evaluation of the system performances in order to explore

the design space. The exploration of design space continues until the system performances can satisfy the requirements.

System-level design tools have been developed to realize system-level design. One of the tools is SystemBuilder. SystemBuilder automatically generates a target implementation from the model according to a mapping decided by designers. The generated implementation can be executed on both simulation tools and Field-Programmable Gate Array. Since SystemBuilder automatically synthesizes the target implementation and the communication interfaces among software and dedicated hardware, it is easy for designers to evaluate system performances of several mappings.

The author first evaluated the design efficiency of SystemBuilder throughout a case study of AES encryption system design. It is clarified that SystemBuilder is effective for designing pipelined systems. In addition, the author unveiled three problems to design systems efficiently, which are 1) limited mapping, 2) long evaluation time, 3) lack of support to improve the system performances. The first problem is limited mapping. Most of system-level design tools can allocate a function to either a software or a hardware module. The tools, however, cannot support different mappings such as several functions share a hardware module. The second problem is long evaluation time. Even though SystemBuilder automatically generates the implementations, it takes very long time for the generation of the implementations and the evaluation of system performances. Thus, with only SystemBuilder, it is hard to evaluate the system performances of a lot of mappings in order to find appropriate mappings. The third problem is lack of support to improve the system performances. SystemBuilder clarifies a bottleneck of execution time by profiling tools. However, it is hard to consider how to improve the system performances of current complex embedded systems with only profiling tools. These are problems of not only SystemBuilder but also the other system-level design tools. Therefore, it is important to solve the problems in order to realize more efficient system-level design.

The author proposes three tools which can overcome the problems above. Three tools are Extended SystemBuilder, Mapping Explorer, and Improvement Analyzer. Extended SystemBuilder is an extension of SystemBuilder, which can generate a hardware module which is shared by several functions. Since Extended SystemBuilder increases the type of mapping, it solves the first problem. Mapping Explorer uses an efficient algorithm named pareto-update search. Since pareto-update search drastically decreases the number of exploration of mappings

to find appropriate mappings. Thus, Mapping Explorer can decrease the time to evaluate the system performances, which is the second problem. Even if Mapping Explorer cannot find an appropriate mapping, Improvement Analyzer helps the designer to identify bottlenecks. In addition, Improvement Analyzer lists several candidates of the ways to improve the system performances so that the designer can consider how to modify the model. Improvement Analyzer, therefore, can overcome the third problem.

Throughout the three tools, exploration of mappings and bottleneck analysis are done automatically to accelerate the design space exploration at system-level. Therefore, system designers can efficiently design large-scale embedded systems with multi-core processors and dedicated hardware.

This dissertation describes the detail of three tools and evaluates them using case studies. In the case studies, efficient design space exploration is demonstrated.