

Title	アプリケーションに適応した組込みプロセッサとリアルタイムOSの自動生成に関する研究
Author(s)	宮内, 哲夫
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	ETD
URL	<a href="http://hdl.handle.net/10119/17480">http://hdl.handle.net/10119/17480</a>
Rights	
Description	Supervisor: 田中 清史, 情報科学研究科, 博士

## Abstract

In recent years, the use of embedded microprocessors has been increasing with prevailing IoT (Internet of Things) and RTOSs are commonly used to develop a real-time system effectively. By using an RTOS, a complicated procedure can be divided into tasks with real-time scheduling based on the preemptive and priority-based task scheduling, and a real-time application can be developed easily with using RTOS functions such as task management, task dependent synchronization, synchronization and communication. While there are advantages for using an RTOS, an RTOS itself consumes additional memory, computational resources and power. Based on these points, our research objectives are as follows: (1) Build an application adaptive processor core. (2) Remove unused codes in RTOS kernel, while leaving necessary functions such as checking possible errors. (3) Implement hardware RTOS to reduce the amount of software resources and execution time. (4) Remove unused codes in a hardware RTOS as well as a software-only RTOS. (5) Build automatic development environment with which we can perform the items above.

In order to achieve the objectives, we adopted MIPS32 architecture for a processor core and illustrated the method for analyzing an application program and generating the application adaptive processor core circuit. In addition, we implemented two- to eight-core multi core processor on an FPGA and showed eight-core processor can be implemented on a relatively small FPGA with application adaptive processor cores.

Regarding RTOS, we proposed a framework to generate application adapted hardware RTOS and software-only RTOS. For the specification of an RTOS, we adopted  $\mu$ ITRON4.0 for the research as it is widely used and its specification is open in public. We propose the methods, “Removing Unnecessary Codes Caused by Fixed Attributes” and “Removing Unnecessary Codes Caused by the Way of Calling”, for generating an application adaptive RTOS kernel. For the former method, as each system call is specified with attributes through parameters in a configuration file, functions which are not specified in the configuration file can be deleted from the RTOS kernel. For the latter method, error codes for system calls are defined in the RTOS specification whereas codes for checking errors which never occur in the application program remain in some cases. Since those codes are redundant when an application program is fixed, it is shown that how unnecessary error checking can be removed. In addition, we explained the structure of the hardware RTOS, which consists of RTOS Hardware Wrapper and RTOS Hardware Core. We propose an environment to generate an application adaptive processor core and a hardware/software-only RTOS kernel in a fully automatic manner.

For the evaluation of the effect of the proposal, we applied the proposed methods to several application programs and measured FPGA resources, RTOS kernel execution time and the size of the software parts. As a result, it can be seen that the hardware resources and the size of a software part of an RTOS kernel are reduced, and that the system call execution time is improved.

*Keyword: processor, MIPS, RTOS,  $\mu$ ITRON, configuration, system call, FPGA, adaptation*