VLSI Implementation of a Real-time Operating System

Takumi Nakano[†], Yoshiki Komatsudaira[††], Akichika Shiomi[‡] and Masaharu Imai[§]
[†] Department of Information and Computer Engineering,
Toyota College of Technology, 2-1 Eisei, Toyota, 471 Japan.
[††] Department of Information and Computer Sciences,
Toyohashi University of Technology, 1-1 Hibariga-oka, Tenpaku, Toyohashi, 441 Japan.
[‡] Department of Computer Science, Faculty of Information,
Shizuoka University, 3-5-1 Johoku, Hamamatu, Shizuoka, 432 Japan.
[§] Department of Computer Science, Graduate School of Engineering Science,
Osaka University, 1-3 Machikane-yama, Toyonaka, Osaka, 560 Japan.
E-mail: sos@arc.tctice.toyota-ct.ac.jp

Abstract— This paper proposes a new approach to realize a very high performance real-time OS using VLSI technology. In order to confirm the effectiveness of this method, the most basic system calls have been designed. According to the evaluation results based on a gate array implementation, hardware portion of system calls can be executed within 4 clocks and the task scheduler can be performed in only 8 clocks simultaneously, which are about 130 to 1880 times faster than software implementation.

I. Introduction

A real-time OS is being used in real-time control systems for various products. In these fields for the embedded systems, highly functional real-time OS and efficient task scheduler are needed. The requirements to the real-time OS can be roughly divided into response time and the worst case execution time. And, these items that have relation to the overhead of the real-time OS must be made to decrease as much as possible. Various approaches to speed-up by software and hardware are shown in Table I.

TABLE I
The approach to speed-up.

Process	Software	Hardware	
		General	Special
Application	improvement of	high	ASIC
Real-time	algorithm and	performance	VLSI
OS	data structure	CPU	ASSP

To improve the performance by software, the improvement of data structure and algorithm is adapted. And, speed-up for the whole process which includes real-time OS and applications has been done by adopting a high performance CPU. Furthermore, ASICs have been used for special applications. But, the speed-up of the real-time OS by VLSI or Application Specific Standard Product (ASSP) has not been implemented yet. And, the behavior of the real-time OS is very different from the application software.

II. The Approach as a VLSI Chip

We have proposed an approach to implement system calls and a scheduler as a peripheral LSI chip [1]. The processing time of system calls and scheduler was decreased as against a conventional implementation. The processing flow of the approach by hardware implementation is shown in Figure 1.

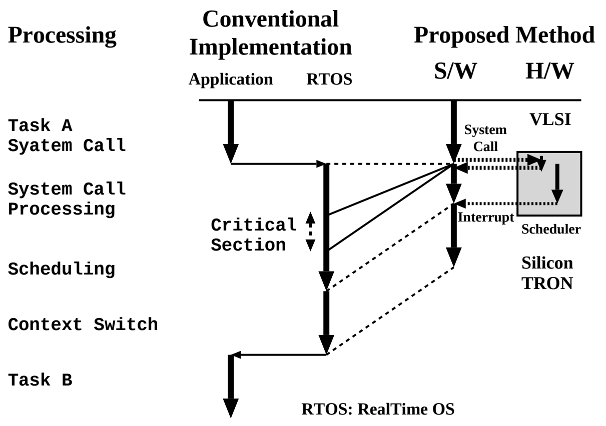

Fig. 1. The processing flow by hardware implementation.

III. The Concept of Silicon OS

The hardware which implements system call functions of μITRON is called a “Silicon TRON” or “Hardware Kernel”. The software which implements other system call functions and interface process between applications and the Silicon TRON is called a "Micro Kernel" or "Software Kernel". And, the whole real-time OS including the Silicon TRON and the micro kernel is called a Silicon OS [2]. The concept of the Silicon OS is shown in Figure 2.

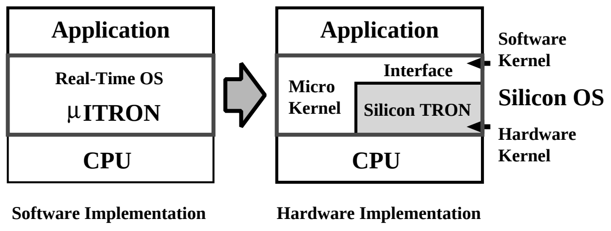

Fig. 2. The concept of Silicon OS.

A. The Silicon TRON

The Silicon TRON implements most effective system calls in hardware to improve the performance of the real-time OS. Other system calls are implemented in software to maintain compatibility and extensibility. And, the Silicon TRON executes a synchronous request from the system call and an asynchronous request from the external event simultaneously in hardware. The Silicon TRON processing flow of system calls and interrupt is shown in Figure 3.

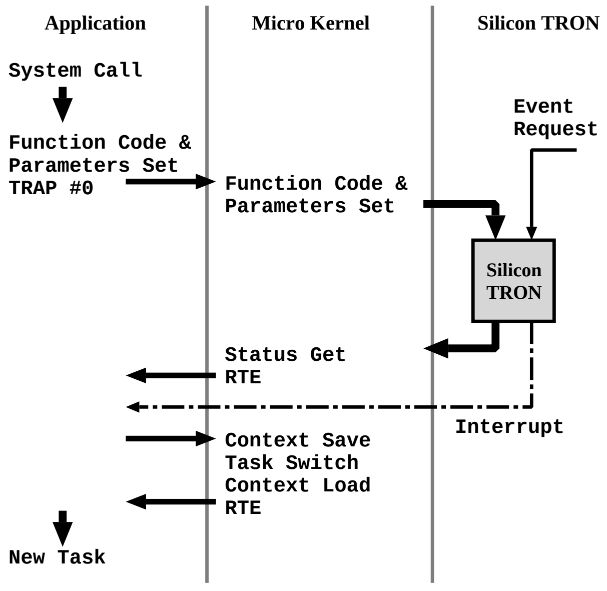
Application
Micro Kernel
Silicon TRON
System Call
Function Code & Parameters Set TRAP #0
Function Code & Parameters Set
Event Request
Silicon TRON
Status Get RTE
Interrupt
Context Save Task Switch Context Load RTE
New Task

Fig. 3. The Silicon TRON processing flow.

B. System Call Processing

The Silicon TRON decodes a system call from the application program and executes it in hardware within a few clocks.

C. External Event Processing

The Silicon TRON changes the internal status at external event such as interrupt signal.

D. Scheduler

The scheduler chooses a task using internal status of tasks by a hardware algorithm. The interrupt signal to execute a context switching is sent to the CPU from the Silicon TRON if the newly chosen task is different from the one currently executed.

IV. Evaluation Result

Part of the system call processing time is shown in Table II.

TABLE II
The system call processing time.

System Call	Processing Clocks		Ratio
	Hardware	Software	
set_flg	2	3760	1880
sem_sts	3	390	130
wai_sem	4	1020	255

The system call processing time of proposed method in hardware can be reduced to within 4 clocks which are 130 to 1880 times faster than the conventional implementation in software. The interrupt disable time of the Silicon TRON has been reduced to within 4 clocks in hardware which is much shorter than that of software implementations. The scheduling process can be performed in only 8 clocks in the Silicon TRON when the number of the tasks is 7. But this scheduling time can be ignored because the hardware scheduler can executes the scheduling concurrently with the micro kernel.

V. Conclusion

The processing time of system calls and the task scheduler can be drastically reduced by implementing a real-time OS in hardware [3]. Accordingly, it is possible to apply the Silicon TRON chip to various application embedded systems such as portable multi-media equipment, mobile robots, avionics and so on.

References

[1] T. Nakano, M. Itabashi, U. Andy, A. Shiomi and M.Imai. The Evaluation of Silicon TRON Design. Proceedings of the RTP '94 (Mar. 1994), pp.79–86. IEICE SIG Notes, CPSY 93–62, in Japanese.

[2] T. Nakano, U. Andy, M. Itabashi, A. Shiomi and M.Imai. VLSI Implementation and Evaluation of a Real-Time Operating System, Transactions of IEICE, Vol. J78–D-I, No.8 (Aug. 1995), pp.679–685, in Japanese.

[3] T. Nakano, U. Andy, M. Itabashi, A. Shiomi and M.Imai. Hardware Implementation of a Real-time Operating System. Proceedings of the Twelfth TRON Project International Symposium, pp. 34–42, IEEE Computer Society Press, Nov. 1995.