

# The KIT COSMOS Processor: Eliminating Ineffectual Branch Instructions via Concurrent Dynamic Optimization\*

Toshiyuki Yamamoto†    Toshinori Sato‡    Itsujiro Arita‡

†Department of Artificial Intelligence  
‡Center for Microelectronic Systems  
Kyushu Institute of Technology  
680-4 Kawazu, Iizuka, 820-8502 Japan  
Tel:+81-948-29-7624 Fax:+81-948-29-7601  
tsato@ai.kyutech.ac.jp

## Abstract

We propose a microprocessor architecture which efficiently utilizes next-generation semiconductor technology. Our proposal based on Simultaneous Multi-Threading (SMT) increases the number of beneficial instructions and thus the functional units work efficiently. As well as application programs, a binary code translator which dynamically optimizes the applications on-the-fly is executed on the SMT. We call this mechanism CONcurrent Dynamic OptimizeR (CONDOR). Based on the CONDOR architecture, performance of the applications are improved as well. This paper presents a preliminary study on dynamic elimination of ineffectual branch instructions.

## 1 Overview of CONDOR

For past decades, advancing semiconductor technology has increased the number of transistors on a single chip. The Semiconductor Industry Association predicted that by 2010, 800M transistors will be integrated on a single chip. Contemporary processors cannot utilize such an unprecedented number of transistors efficiently, because increasing the number of functional units cannot contribute to processor performance due to low instruction level parallelism (ILP) in programs such as desktop applications. In order to extract more ILP, modern processors rely on speculative execution or multi-path execution. However, they extract ILP at the risk of execution of useless instructions and thus they are justified only under the condition where a number of functional units are idle.

These considerations lead us to explore a new way to extract beneficial instructions. Idling functional units are efficiently utilized by multi-threading. An application and a optimizing program execute simultaneously on a multi-threaded processor. The opti-

mizing program improves performance of the application on-the-fly. We call this technique CONDOR [1,2]. This execution model is similar with SMT. Therefore, we investigate a processor architecture utilizing CONDOR on the top of the SMT processor, which we call COSMOS processor.

## 2 Eliminating Ineffectual Branches

CONDOR is a general purpose instruction optimizer, hence it can do traditional optimization methods as modern compilers do. In addition, CONDOR can perform aggressive optimization for branch instructions as follows. Some branch instructions are highly biased to one direction. Such branches can be replaced by unconditional jumps or be removed completely. This improves utilization of branch prediction tables, increasing branch prediction accuracy. In addition, if branches are removed, basic block size are enlarged. These contribute to exploiting ILP. We evaluate this optimization on `300.twolf` from SPECint2000 benchmark. Note that this optimization is performed by hand. The simulation results is presented at Table 1. We can easily observe, performance of `300.twolf` is improved by 8.41%, and thus confirm the elimination of ineffectual branch instructions via concurrent dynamic optimization is effective.

	# of branches	# of cycles	ILP
Before	32,975,978	192,312,206	1.3455
After	31,269,524	177,390,682	1.4587
%Changes	-5.17%	-7.76%	+8.41%

Table 1: Simulation results

## References

- [1] Sato T. and Arita I.: The KIT COSMOS processor: background and rationale, IEICE CPSY99-115 (2000) in Japanese.
- [2] Yamamoto T., Sato T, and Arita I.: The KIT COSMOS processor: preliminary study on characterizing hot spots, IEICE CPSY2000-61 (2000) in Japanese.

\*This work is supported in part by the Grant-in-Aid for Scientific Research #12780273 from Japan Society for the Promotion of Science, and by the grant No.H12-1 from Fukuoka Industry, Science & Technology Foundation.