

## HARDWARE-SOFTWARE PARTITIONING FOR EMBEDDED MULTIPROCESSOR FPGA SYSTEMS

TRONG-YEN LEE<sup>1</sup>, YANG-HSIN FAN<sup>1,2</sup>, YU-MIN CHENG<sup>1</sup> AND CHIA-CHUN TSAI<sup>3</sup>

<sup>1</sup>Graduate Institute of Computer and Communication  
National Taipei University of Technology  
Taipei 10643, Taiwan  
tylee@ntut.edu.tw; yhfan@nttu.edu.tw; nick.cheng@chroma.com.tw

<sup>2</sup>Department of Computer Science and Information Engineering  
National Taitung University  
Taitung 95052, Taiwan

<sup>3</sup>Department of Computer Science and Information Engineering  
Nanhua University  
Chia-Yi 62248, Taiwan  
chun@mail.nhu.edu.tw

Received April 2008; revised September 2008

**ABSTRACT.** *This work proposes a hardware-software partitioning approach named GHO to solve the partitioning issue for embedded multiprocessor FPGA systems. GHO adopts genetic algorithm and hardware-oriented partition to improve the partitioning result with faster execution time, smaller memory size and higher slice usage under satisfied system constraints. Two experimental results demonstrate that GHO is feasible for solving the hardware-software partition for embedded multiprocessor FPGA systems.*

**Keywords:** Hardware-software partitioning, Genetic algorithm, Hardware-oriented partitioning, Embedded multiprocessor, FPGA system

1. **Introduction.** Nano-meter technology has been increasingly applied to the VLSI fabrication process since the development of deep sub-micron technology. System-on-a-chip (SoC) based products have recently been designed with the advantages of low cost, low power and high performance. Meanwhile, new fabrication process has significantly raised the transistor capacity that can be used to design multiprocessor FPGA systems. The Xilinx [1] Virtex series field programmable gate array (FPGA) is an example of a multiprocessor platform that is suitable for quickly developing highly complex designs. Multiprocessor FPGA systems are becoming increasingly popular for the three reasons. First, they handle floating operations easier than hardware architectures. Second, multiprocessor FPGA systems are more flexible than hardware system in terms of the computing mass and complex data, because they run software in parallel as opposed to hardware. Furthermore, it has more powerful computation for providing real-time services or federated embedded systems.

The hardware-software partitioning phase of developing a FPGA system traditionally depended mostly on the experiences of engineers [2,3]. System integration is thus seen a major challenge, because high-level system analysis and model, design for hardware and software are developed separately. Additionally, the system performance and functionality may be limited or insufficient, since the system of hardware and software may not be fully integrated. Moreover, the partitioning result may not be the best solution in terms of memory size, execution time and efficient utilization on FPGA resources.