


学位論文の要旨

| | | | |
|--|-----------|------------|--|
| 専攻名 | システム工学 専攻 | ふりがな 氏名 | 中林 智之 ㊟ |
| 学位論文題目 Researches on fabrication of low-energy heterogeneous multi-core processors (低電力ヘテロジニアスマルチコアプロセッサの設計に関する研究) | | | |
| <p>Since energy consumption and heat density are growing problems in high-performance processors as well as embedded processors, lots of latest researches on computer systems aim at enhancing energy-efficiency of processors. One of leading energy-efficient approaches, single-instruction-set-architecture (single-ISA) heterogeneous multi-core processor comprised of microarchitecturally diverse cores (differently-designed in microarchitecture level), gets much attention from the researchers. Two key challenges in the heterogeneous paradigm are (1) the development of energy-efficient processor core highlighting finer heterogeneity in an application phase, and (2) the design automation of the entire single-ISA heterogeneous multi-core processor. The author studies basic circuit and architecture for (1), and develops a processor design environment for (2) as described below in detail:</p> <p>(1) This work proposes a combinational approach across two different fields to develop a low-energy processor core, i.e., <i>circuit-level</i> (low-energy D-flip-flops) and <i>microarchitecture-level</i> (variable stages pipeline) approaches.</p> <p><i>Circuit-level approach:</i> D-flip-flops play an important role in a processor chip because the delay, area, and power consumption of D-flip-flops drastically affect the performance of the processor. This work proposes two types of novel D-flip-flop which adopt semi-static and true-single-phase clock (TSPC) schemes. One is called double split-output semi-static TSPC D-flip-flop (DSSTSPC D-flip-flop) emphasizing short circuit delay by a novel front-end composed of parallelized split-output latches. The other, single split-output semi-static TSPC D-flip-flop (SSSTSPC D-flip-flop), takes a special focus on low-energy operation by removing a part of DSSTSPC D-flip-flop. The former shortens the circuit delay by 5% compared with a conventional low-energy D-flip-flop without increase in the energy and layout area. The latter achieves 31% smaller layout area and 30% lower energy consumption with up to 8% performance degradation compared with the conventional D-flip-flop.</p> <p><i>Microarchitecture-level approach:</i> Modern processors widely employ dynamic voltage and frequency scaling (DVFS) technique which dynamically scales the supply voltage and clock frequency in accordance with workload on the processor. Although DVFS is effective for energy saving, it suffers from its large overhead when we intend a temporally fine-grain energy optimization. To compensate for DVFS, a variable stages pipeline (VSP) architecture is proposed. VSP reduces the energy consumption by dynamically varying the pipeline depth, instead of the supply voltage, depending on instruction-level behavior in a running program. Since the penalty for a pipeline scaling is small enough to reduce the energy consumption at tens or hundreds clock cycles, VSP can save the energy consumption at finer-grain period than DVFS. This thesis proposes a fine-grain depth-changing method which can be implemented by a trivial FIFO buffer to detect processor workload, and presents its chip fabrication on a 180 nm technology. Evaluation results using the fabricated VSP chip show that the VSP reduces the energy consumption by 34% to 48% at fine-grained low-energy operation insertion which is impossible with DVFS. Moreover, we adopt a special cell called latch D-flip-flop selector-cell (LDS-cell) into VSP processor to further reduce the energy consumption under folded pipeline structure. This thesis reveals that inserting LDS-cells makes VSP processor consume 13% less energy on a fabricated chip.</p> | | | |

続紙 有 無

| | |
|------------|---|
| ふりがな 氏名 | 中林 智之  |
|------------|---|

(2) This thesis also presents a development environment that improves research productivity by *automatic design generation* and *co-simulation framework*, especially fabrication and prototyping through a standard ASIC design flows.

Automatic design generation. Because a single-ISA heterogeneous multi-core consists of microarchitecturally diverse cores to streamline the execution of diverse program phases, the design and verification effort is multiplied by the number of employed core types. The increased design effort impedes development of heterogeneous multi-core processors. N. K. Choudhary et al. develop a toolset, called FabScalar, for automatically composing the synthesizable designs of arbitrary cores. Although using FabScalar helps mitigate the design effort, the design effort for diverse cache systems and a shared bus still exists as a barrier in the development of heterogeneous multi-core processor. This work proposes FabHetero which is composed of three design automation tools: FabScalar, FabCache, and FabBus for automatically composing diverse cores, cache systems, and flexible shared bus, respectively. FabHetero project sets a goal of fabricating heterogeneous multi-core processor chips in a short time, and this work is the first attempt to automate the entire heterogeneous multi-core design. The author confines the microarchitectural diversity into a superset code that enables users to use a single universal design of heterogeneous multi-core processor; however, the footprint of each design is the desired configuration. FabCache automatically designs many caches that satisfy the requirements from modern superscalars and differ in cache dimensions. FabBus automates generating a flexible shared bus which connects the arbitrary number of caches with desired cache coherence protocol.

Co-simulation framework. Furthermore, FabHetero framework includes a practical processor co-simulation framework for not only RTL simulation but also gate/transistor level simulation, and even fabricated chip evaluation/validation. Our framework addresses the following two challenges: system call emulation and sampled execution. Both mechanisms are commonly used only in software processor simulators; therefore, this work introduces these mechanisms into standard ASIC design flows using off-chip system call emulator and checkpoint mechanism. Processor design can remain unchanged from its pure specification (no extra I/Os and hardware is needed) because the proposed mechanisms exploit general instructions inherent in processor.

This work provides a great step: automatic generation of an entire processor design involving a superscalar core, cache system, and bus system and its fabrication in shortened design time using the co-simulation framework. This helps researchers fabricate their novel processor chips by much less effort.