

Abstract of the Dissertation

Functional Partitioning for Low Power

by

Enoch Oi-Kee Hwang

Doctor of Philosophy, Graduate Program in Computer Science

University of California, Riverside, June, 1999

Dr. Frank Vahid, Chairperson

Dr. Yu-Chin Hsu, Co-Chairperson

*Power reductions in VLSI systems have recently become a critical metric for design evaluation. Although power reduction techniques can be applied at every level of design abstraction, most automated power reduction techniques apply to the lower levels of design abstraction. Previous works have shown that sizable power reductions can be achieved simply by shutting down a system's sub-circuits when they are not needed. However, these shutdown techniques focus on shutting down only portions of the controller or the datapath of a single custom hardware processor. We therefore investigated the power reduction attainable by the evolving automated technique of functional partitioning in which a process is automatically divided into multiple simpler, mutually exclusive, communicating processors, and then shut down the inactive processors. By shutting down the entire inactive processor, we have in effect shut down both the controller and datapath. Power reduction is accomplished because only one smaller processor is active at a time.*

*We have applied this functional partitioning technique to either the procedural or the finite-state machine with datapath behavioral level. From either level, the original process is partitioned into multiple parts. For the procedural level, a coarse-grained partitioning of procedures is done. Data transfers between the parts are simply the parameters in the procedural call. In contrast, FSM partitioning has no concept of procedures, but rather states. A dataflow analysis is first performed to determine the data transfers between the parts. A power partitioning algorithm is then used to separate the states into multiple parts. The parts are then individually synthesized down to the gate level netlist. Finally, communication is added between the parts so that they are functionally equivalent to the original unpartitioned process.*

*Partitioning introduces extra power consumption for inter-processor communication. Thus, the problem that must be solved is one of partitioning such that the reduction in power for computations far outweighs the power increase for communication, while also minimizing the increase in total circuit size and execution time. Our results show that this functional partitioning technique can reduce power, on average, by 42% over unoptimized systems. In addition to power reduction, functional partitioning also provides solutions to a variety of synthesis problems.*