Memory Optimization Techniques for Low-Power Embedded Processors

Manish Verma, Peter Marwedel

Department of Computer Science XII
University of Dortmund
Otto-Hahn Strasse 16
44221 Dortmund
{manish.verma,peter.marwedel}@uni-dortmund.de

Power consumption is an important design issue for contemporary portable embedded devices. It is known that the next generation of portable devices will feature faster processors and larger memories, both of which require high operational power. Memory subsystem has already been identified as the energy bottleneck of the entire system. Consequently, memory hierarchies are being constructed to reduce the memory subsystem's energy dissipation. Caches and scratchpad memories represent two contrasting memory architectures. Scratchpads are both area and power efficient than caches. However, they require explicit support from the compiler for managing their contents. In this work, we present three approaches for the prudent utilization of the scratchpad memory of an ARM7 processor and of a M5 DSP based system.

The first approach is based on the following observations. Firstly, a small memory requires less energy per access than that by a large memory. Secondly, applications in general consist of small and frequently accessed arrays and large but infrequently accessed arrays. Consequently, the approach partitions the large scratchpad into several small scratchpads. The arrays are also statically mapped such that the small arrays are mapped to small and energy efficient scratchpads. The approach leads to average energy savings of 52% and 35% in the data memory subsystem of the ARM7 and the M5 DSP, respectively.

The second approach utilizes the scratchpad as an instruction buffer in a cache based memory hierarchy. The approach models the cache as a conflict graph and assigns instructions to the scratchpad. The objective is to minimize the energy consumption of the system while preserving the predictable behavior of the memory hierarchy. The approach results in an average energy saving of 21% against the above approach for the ARM7 based system.

The last approach optimizes the energy consumption of the system by overlaying memory objects (*i.e.* code segments and data elements) on to the scratchpad. Memory objects with non-conflicting life-times are assigned to the same location on the scratchpad. This improves the scratchpad utilization, however, it requires copying memory objects on and off the scratchpad during the execution of the application. Average energy reductions of 34% and 33% are reported for the ARM7 and the M5 DSP based systems, respectively.