AN ENERGY-AWARE EXPLORATION APPROACH BASED ON OPEN SOFTWARE ENVIRONMENT

A.G. Silva-Filho, R. Eskinazi, P.S.B. Nascimento, and M.E. Lima
Federal University of Pernambuco¹ and Politechnic School of Engineering²
Fone: (+55) 8121268430, Recife-PE, Brazil
{agsf, res, psbn, mel}@cin.ufpe.br

Abstract: In this work it is presented an automated method for tuning memory hierarchy to embedded applications in order to reduce energy consumption. Detailed studies were performed on TCaT heuristic in order to obtain a new exploration approach. Experiments show a reduction of approximately 4 times in the energy consumption by using our heuristic, considering two-level caches. An open software environment, based on SystemC, to explore architectures aiming energy consumption on cache memory hierarchy, has been extended. MediaBench benchmark has been used.

Keywords: Energy Consumption, ADLs, Exploration Heuristics

1. INTRODUCTION

Nowadays, cache memory structure may consume up to 50% of a microprocessor's energy [1]. This situation motivates designers of embedded microprocessor platforms to evaluate the impact of such parameter in their applications regarding to performance, cost and energy usage [3].

Among existing Architecture Description Language (ADLs), ArchC [2] is offered as a promising one based on SystemC, able to model processors, as well as memory hierarchies. Although ArchC does not extract energy consumption results from architecture behavior, its advantage is that it allows a previous analysis of the architecture, before hardware synthesis.

The traditional method to explore the huge solution space of possible configurations for a given memory hierarchy is based on exhaustive searching. However, it is often prohibitive due to the long spent time.

The heuristic developed by Zhang et al. [6] is based on single cache level approach. Two out of three parameters are maintained fixed during analysis in order to study the impact of the third one on the miss rate and energy consumption.

Although the previous heuristic was intended for only one level, Gordon-Ross [7] applied this heuristic and observed that for two cache levels the results were not good. Thus, Gordon-Ross extended this “initial heuristic” and proposed the TcaT heuristic that provides energy savings of 53% when compared with Zhang heuristic.

Summary: In this work we propose two contributions: (i) An ArchC extension to provide energy consumption evaluation, and (ii) A new heuristic for tuning memory hierarchy to embedded applications considering two-levels cache, intended for reducing energy consumption.

2. ENERGY-AWARE ARCHC MODEL

Currently, the ArchC [2] ADL does not allow the architecture exploration for platform-tuning in terms of energy consumption. Our work extends ArchC to allow the accomplishment of this.

The ArchC pre-processor takes as input the system-level specification, including the instruction set and architecture description, and automatically generates a SystemC behavioral model of the architecture. The SystemC model is compiled with GCC and it generates an executable specification of the architecture. By executing this model, reports about the system behavior are issued.

Some parameters such as: supply operation and technology parameters were incorporated to the ArchC description in order to estimate energy consumption.

A set of classes were integrated to the ArchC language, allowing its use in energy-aware descriptions of memory hierarchies for processor-based architectures. Each instance defined in the AC_ARCH description is monitored through “*read*” and “*write*” behavioral operations. All these accesses are annotated and so the energy consumption is calculated. The outcome from the generated model simulation is the consumed energy for each cache component described.

There are basically two components that establish the amount of energy dissipated in a CMOS circuit: Static and dynamic dissipations. In this work, the two components are modelled. The equations of the eCACTI power analytical model [5] were chosen to model the cache memory energy

consumption, since it offers results at a reasonable accuracy for high-level descriptions.

3. PROPOSED HEURISTIC

TECH (Two-level Cache Exploration Heuristic) allows the exploration of the two-level caches and selects the cache parameters in a specific order based on their impact on energy. First, the best cache size is obtained for the level two cache by varying its size decreasingly. That is made varying only the cache sizes of the second level from a maximum to a minimum value, which can't be inferior to the first level cache size. The procedure stops when the energy consumption of the new cache configuration is bigger than a previous configuration. Once found the best configuration for the second cache level (lesser energy consumption), this value is fixed, and the exploration continues varying increasingly the cache size of first level until to find the best configuration. This configuration is found varying only the cache size of the first level from a minimum value to a cache size immediately smaller than the second level. The stop criterion is the same as in the second level. Here, the first and second cache sizes are fixed. Now, in a second stage, the heuristic increases the cache line size of the second level initially, until find a new configuration with energy consumption inferior to previous configuration. The same procedure is carried out with cache of first level. Finally, the same procedure is carried out with cache associativity, for the two cache levels.

4. SPECIFYING THE SYSTEM

In Figure 1, a MIPS memory hierarchy is composed by separate level one instruction (IC1) and data caches (DM1), separate level two instruction (IC2) and data caches (DM2) and main memory (MEM). IC1 is declared as a direct-mapped, 1024 blocks (or lines), 2 words/block, write-through, write-allocate, 1.7V power supply, 500MHz frequency cache and 0.08 μ m transistor technology. DM1 is declared as a two-way set associative, 2048 lines, 1 words/block, LRU replaced, write-back, write-allocate, 1.7V power supply, 500MHz frequency cache and 0.08 μ m transistor technology. IC2 and DM2 are declared of a similar manner. In ArchC, the method *bindsTo* defines how the memory hierarchy is established.

```

AC_ARCH(mips1) {
  ac_mem MEM:5M;
  ac_powercache IC1("dm", 1024, 2, "wt", "wal", 1.7, 500, 0.08);
  ac_powercache IC2("dm", 2048, 2, "wt", "wal", 1.7, 500, 0.08);
  ac_powercache DM1("2w", 2048, 1, "lru", "wb", "wal", 1.7, 500, 0.08);
  ac_powercache DM2("2w", 4096, 1, "lru", "wb", "wal", 1.7, 500, 0.08);
  ac_regbank RG:8; ac_regbank RB:256; ac_reg PSR,Y;
  ARCH_CTOR ( mips1 ) {
    ac_isa("mips1_isa.ac");
    IC1.bindsTo( IC2 );
    DM1.bindsTo( DM2 );
    IC2.bindsTo( MEM );
    DM2.bindsTo( MEM );
  };
};

```

Figure 1. Mips Memory hierarchy

5. CASE STUDY

Here an example using a representative audio compression algorithm (ADPCM) from MediaBench suite [4] is applied. A Mips core processor with memory hierarchy structure was used. It was considered a power supply of 1.7V, write-back and write-allocate write policy, 500MHz cache frequency and 0.08 μ m transistor technology.

The proposed heuristic (TECH) can be applied separately for instruction and data caches. In order to demonstrate the heuristic evaluation, only instruction cache is analysed. We apply the proposed heuristic using the proposed environment (ArchC). The cache range parameters in this case was: 2k to 512k bytes for cache size, 8 to 64 bytes for cache line size and 1 to 8 for associativity.

Studies done over this application indicate that the energy consumption contributions by varying of the cache line size and cache associativity are lower than the contributions due the varying of the cache size. Based on this studies, we discover by simulations that to define the cache size of the last cache level, initially, is important to limit the increases of first cache level.

Preliminary studies have shown that our heuristic has reached results as good as TcaT in this case. The results for this application concludes that different order exploration can conduce the cache memory energy consumption for different levels. Figure 2 shows the normalized energy consumption, for level one and two caches considering both heuristics TcaT (case 1) and TECH (case 2).

Figure 2. Different energy levels (two cases)

In this case, results can demonstrate that some care must be done when architecture space exploration viewing energy consumption is performed. TCaT [7] uses a configurable cache approach (by using reconfigurable device such as FPGAs) with limitted exploration space due to the fact that cache architecture only supports a certain range of configurations. This aspect can discard importants configurations that consumes lower energy in the space exploration. On the other hand, by using our heuristic approach, we are worried with the cache configuration that provide low energy consumption. Experiments show a reduction of approximately 4 times in the energy consumption. For this case, our results are as good as TCaT [7] results.

6. CONCLUSION

An environment capable to extract cache memory energy consumption from an architecture behavior has been suggested. This extension enables a suitable evaluation at system-level specification. Through energy consumption reporting for each cache memory component, the designer is able to analyze and adjust the efficiency of the architectural parameters that may impact on the overall energy consumption and performance of the system.

A heuristic (TECH) for tuning memory hierarchy to embedded applications in order to reduce energy consumption is presented. The proposed heuristic have presented good results for two-level caches energy analyses, when compared with existing heuristic. Preliminary results, show a reduction of approximately 4 times in the energy consumption.

Additionally, a cycle-accurate power cache model is under development to provide a more detailed power analysis.

REFERENCES

1. H. Chang; L. Code; M. Hunt, G. Martin, A.J. McNelly, and L. Todd, "Surviving the SOC revolution: a guide to platform-based design"; Kluwer A.P., 1 ed., 1999.
2. The ArchC Architecture Description Language, 2004; In: www.archc.org
3. Macii, Enrico; L. Benini; M. Poncino; "Energy-Aware Design of Embedded Memories: A Survey of Technologies, Architectures and Optimization Techniques", *ACM Transactions on Embedded Computing Systems*; Vol. 2, No. 1, pp. 5-32, February 2003.
4. MediaBench. The MediaBench benchmark suite. In <http://cares.icsl.ucla.edu/MediaBench/>, 2004.
5. Dutt, Nikil; Mamidipaka, Mahesh; "eCACTI: An Enhanced Power Estimation Model for On-chip Caches", TR 04-28; set. 2004.
6. Zhang, D., Vahid, F., Cache configuration exploration on prototyping platforms. 14th IEEE International Workshop on Rapid System Prototyping, June 2003.
7. Gordon-Ross, Ann, Vahid, F., Dutt, Nikil, Automatic Tuning of Two-Level Caches to Embedded Applications, DATE, pp.208-213, Feb 2004.