Modeling and Simulating Memory Hierarchies in a Platform-based Design Methodology

Pablo Viana*, Edna Barros
CIn - UFPE
Recife-PE, Brazil
{pvs, ensb}@cin.ufpe.br

Sandro Rigo†, Rodolfo Azevedo, Guido Araújo
IC-UNICAMP
Campinas-SP, Brazil
{srigo, rodolfo, guido}@ic.unicamp.br

Abstract

This paper presents an environment based on SystemC for architecture specification of programmable systems. Making use of the new architecture description language ArchC, able to capture the processor description as well as the memory subsystem configuration, this environment offers support for system-level specification, intended for platform-based design. As a case study, it is presented the memory architecture exploration for a simple image processing application, yet a more robust environment evaluation is performed through the execution of some real-world benchmarks.

1. Introduction

Platform-Based Design methodology [4] have been adopted for the development of digital systems. Supported by such a platform, the digital system design may be started from an architecture description where the application's behavior can be mapped on. According to the system requirements and constraints, the system modules are tuned toward the desired performance and cost. Able to capture the entire programmable system specification, ArchC [6] is presented as a promising architecture description language (ADL), ready to model the main features of the processor, as well as the memory subsystem. Over such a model, it is possible to map a software application, and the simulation of the software running on the architectural model can help the designer to evaluate its efficiency and to adjust the parameters that may impact on the overall performance and power consumption of the system [3]. In this paper, it is proposed an environment to model and to simulate storage devices in operation with a processor architecture.

* Partially supported by UFAL
† Supported by FAPESP(00/14376-2)

2. An Environment for System Specification

Figure 1 illustrates our environment for modeling and exploration of the architectural design space. The pre-processor takes the system-level specification and automatically generates a SystemC behavioral model of the architecture. The code generated is compiled, resulting in an executable specification of the architecture. When executed, the simulator produces selective reports about the system performance, such as number of cycles, cache miss rate, data transferred, and all relevant information useful for architecture tuning. Aiming to improve application performance or decrease hardware cost, the designer may alter the processor's instruction set, memory hierarchy or cache parameters and then immediately simulate the system to evaluate the new configuration.

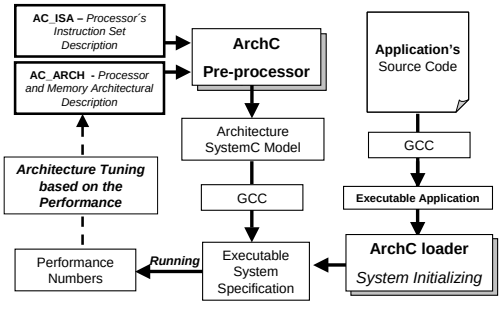

Figure 1. Environment overview.

ArchC is an architecture description language initially conceived for processor architecture description with enough expression power to model several classes of architectures. This work extends the language in order to allow a more detailed memory subsystem description composed by caches and memories. An ArchC description is divided in two parts. The `Instruction Set`

`Architecture (AC_ISA)` description provides instruction formats, size, names, decoding info and instruction behaviors. The `Architecture Resources (AC_ARCH)` description comprises storage devices, pipeline stages and memory structure. Based on these two descriptions, ArchC can generate a behavioral SystemC simulator for the architecture. For this work, we have adopted the Leon [2] (Sparc-V8) architecture to compose our system. Figure 2 shows its architecture resources description (AC_ARCH), exemplifying the cache declarations. A more detailed description of ArchC and its resources can be found in [1, 6].

```
AC_ARCH(leon){

  ac_icache   IC("dm", 128, "wt", "war");
  ac_dcache   DC("2w", 64, 4, "lru", "wt","war");
  ac_mem   MEM:256K;
  ac_regbank RG:8;
  ac_regbank RB:520;
  ac_reg PRS, Y, WIM;

  ARCH_CTOR(leon){
    ac_isa("leon_isa.ac");
    icache.bindTo( MEM );  //Memory hierarchy
    dcache.bindTo( MEM );  //construction
  };
};
```

Figure 2. Memory hierarchy in ArchC.

3. A Case Study and Experimental Results

As an illustrative example consider a convolution algorithm, often used in image processing applications, running on a Leon-based architecture. The model of our platform consists of a processor and a memory hierarchy composed by instruction and data caches connected to a main memory. Table 1 presents performance results of 12 different configurations with varying cache sizes, grouped into: A)Direct-mapped; B) Multi-word blocks; C) Set-associative and D)Combined.

	Data configuration		Read miss rate
A	32-blocks	1-word	43.29%
	64-blocks	1-word	23.31%
	256-blocks	1-word	19.30%
B	32-blocks	2-words	27.54%
	32-blocks	4-words	22.01%
	128-blocks	2-words	23.92%
	128-blocks	4-words	17.61%
C	128-blocks	2-way	19.94%
	128-blocks	4-way	36.88%
	256-blocks	2-way	18.84%
	256-blocks	4-way	18.79%
D	64-blocks	4-words, 2-way	17.54%

Table 1. Data miss rate

The architecture description was also utilized for the execution of some benchmarks from the MiBench suite [5]. Figure 3 shows the hit rate evaluation of these programs, for some cache configurations: 64kB direct-mapped (Config1); 256kB, 2words/block, 4-way (Config2); 8kB, direct-mapped (Config3); 16kB, 8-way (Config4).

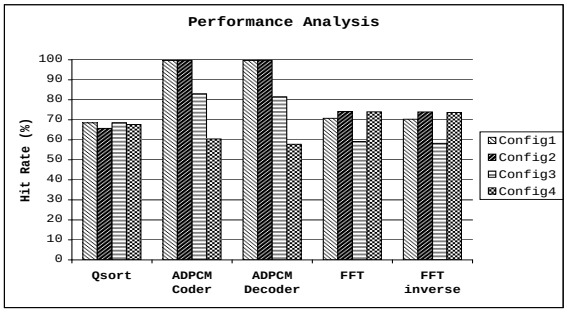

Figure 3. Performance of media benchmarks.

4. Conclusions

Intended for platform tuning, the ArchC environment enables the designer to rapidly simulate the system-level specification and get performance results for a running application. The metrics obtained can help the adjustment of the architecture elements, mainly the memory hierarchy.

In fact, the environment here presented constitutes the platform for the development of a methodological memory hierarchy tuning for embedded applications.

As a SystemC based language, the model generated by ArchC permits the addition of elements that do not compose the processor but are present on chip, like buses and peripherals.

References

[1] ArchC Website: http://www.archc.org.

[2] J. Gaisler. *The LEON Processor User's Manual Version 1.0.10*. Gaisler Research, January 2003.

[3] T. Givardis, F. Vahid, and J. Henkel. System-level exploration for pareto-optimal configuration in parameterized system-on-a-chip. *IEEE Trans. on VLSI Systems*, 2002.

[4] H. Chang et al. *Surviving the SOC Revolution: A Guide to Platform-Based Design*. Kluwer Academic Publishers, 1999.

[5] M. R. Guthaus et al. MiBench: A free, commercially representative embedded benchmark suite. In *Proc. 4th IEEE Annual Workshop on Workload Characterization*, December 2001.

[6] P. Viana, E. Barros, S. Rigo, R. Azevedo, and G. Araújo. Exploring Memory Hierarchy with ArchC. In *Proc. of the 15th Symp. on Computer Architecture and High Performance Computing (SBAC-PAD)*, November 2003.