Standardizing the Performance Assessment of Reconfigurable Processor Architectures[†]

Lesley Shannon and Paul Chow
Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON, Canada, M5S 3G4
{lesley, pc} @eecg.toronto.edu

Comparing and evaluating the performance of reconfigurable processors is a difficult task. For this research field to progress in a more meaningful and scientific fashion, there needs to be a method of measuring progress between different reconfigurable architectures as well as with respect to traditional computing technologies.

This paper presents the Reconfigurable Architecture TEsting Suite, or RATES, which defines a standard for describing and using benchmarks for reconfigurable architectures. RATES is a set of functional benchmarks, is totally independent from the architecture and language, and usable on any processing platform be it general purpose or reconfigurable. It requires standard algorithms to allow comparisons amongst architectures but allows custom algorithms to highlight specific features.

Although creating a standard set of benchmarks seems to be an obvious solution, there are many issues to be addressed if this is to be realized. The first is that there is no *standard* language for algorithmic abstraction that can be compiled to all reconfigurable processors. This is partially due to designers choosing different computational models for their processors, and thus adopting whatever language model enables a reasonable mapping to their architecture. RATES addresses this problem by abstracting the benchmark problem definition from the source code.

A second problem arises from the fact that there aren't any standard benchmarks for these processors. Thus researchers may choose different algorithms when trying to implement the same benchmark to ameliorate performance. However, if a researcher does not use the same algorithm to implement the same benchmark there is some question as to the usefulness of the results for interproject comparisons.

Problems also arise from considering what types of tests should be included in a reconfigurable benchmark suite. Most projects use numerous kernels to test their reconfigurable fabric and only a few applications to measure the overall performance of the processor. However, kernels do not provide a complete measure of a processor's performance. Furthermore, benchmarks should represent both control and data intensive applications to expose weaknesses in the overall design, such as overheads in the host interfaces, memory, and configuration. Ideally, RATES will contain a balance of kernels and applications representing both control and data intensive programs.

As can be seen from the issues outlined in the above discussion, defining a standard benchmark suite for reconfigurable architectures is not a trivial task. The objective is for RATES to be a suite of benchmarks that is usable by any processor designer, independent of the underlying architecture. By standardizing the benchmarks, the comparisons among different architectures can be facilitated. This ensures a more scientific methodology for measuring performance and highlights the benefits and challenges of reconfigurable architecture design.

The first work attempting to address the need for benchmarks for reconfigurable processors was done by S. Kumar et al. [1] at Honeywell Technology Center. Honeywell's suite also uses a benchmark specification format independent of language implementation similar to RATES. However, the emphasis of their *stressmarks* is to evaluate specific characteristics of an architecture and its corresponding infrastructure rather than performance. Thus the goals of the Honeywell suite and RATES are actually complementary.

The Network Processing Forum (NPF) Benchmarking Work Group wrote a white paper [2] also presenting a new format for network processor benchmarks. They face a problem similar to the reconfigurable community in that, by definition, they include a variety of different architectures and physical implementations. To address this problem, they have proposed that each benchmark be specified using a functional description, as opposed to a source code description [2]. They have not outlined a standardized for-

[†]This research was supported by the Natural Sciences and Engineering Research Council.

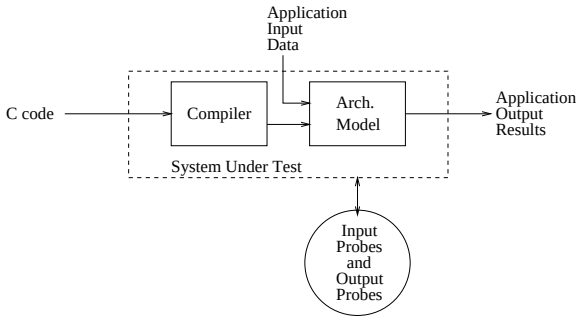

Figure 1. The traditional format for testing processor architectures that use a C code compiler.

mat for describing applications, whereas RATES includes a set of *rules* that must be followed when specifying a benchmark.

Figure 1 illustrates the traditional method used to measure architecture performance where the dotted box defines the system being tested. As shown in the figure, the system under test includes both the compiler and the hardware model. Note that the terms *application input data* and *application output results* are used to delineate the inputs and outputs for the actual application being executed. In contrast, the *probe inputs* and *probe outputs* are used to measure the performance and resource usage of the processor.

For this benchmarking structure, both the application input data and source code are provided. Furthermore, the application output results are defined to guarantee that there are no errors in the compilation process or the processor's execution. While it is possible to provide reconfigurable architects with application input and output data for each benchmark, the traditional benchmark model must be adapted to address the lack of source code standards for reconfigurable processors. The benchmarking format proposed in Figure 2 accomplishes this goal by moving to a higher level of abstraction and providing an algorithmic representation of the benchmark. As can be seen from the figure, the designer must create the native source code that will be recognized by the system's compiler.

From the new structure outlined in Figure 2, it is apparent that the system being measured can accept one of two algorithm types. The first is the *Standard Algorithm* that is specified as part of the benchmark suite. The second is what is being termed as the *Custom Algorithm* that allows the user to restructure the algorithm to utilize architectural features specific to that processor to improve the performance of the benchmark. It is essential to note that the results for the Standard Algorithm are required to relate the results between Custom and Standard Algorithms among all architectures measured using this methodology.

However, a description of the Standard Algorithm is only a portion of a complete RATES benchmark specification,

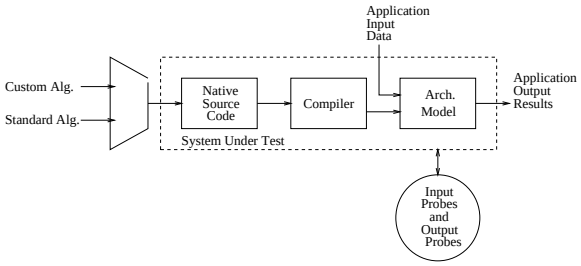

Figure 2. A new format for testing reconfigurable architectures.

which consists of:

- The Standard Algorithm Specification.
- The Application Input and Output Data Files.
- The C code Algorithm Description. This source code is structured to clearly describe the Standard Algorithm rather than to achieve performance.
- The C code Efficient Implementation. This may be necessary if the source code describing the algorithm has poor run time perfomance. In this case, the optimized C code should be used for all performance measures and comparisons.
- The README file. It provides the user with a description of how to execute the C examples, the input and output file formats, and any other information specific to this benchmark.

For a detailed description of the Standard Algorithm specification, the benchmark specification guidelines and further information on RATES, please see: *http://www.eecg.utoronto.ca/˜lesley/benchmarks/rates/*. Benchmarks, input and output data sets, and sample C code can be downloaded from this website. To date, the only benchmarks included in the suite are the 2D-DCT and John Conway's Game of Life. Input from the research community is sought for help to expand the suite and ensure that it addresses the important design issues.

References

[1] S. Kumar, L. Pires, S. Ponnuswamy, C. Nanavati, J. Golusky, M. Vojta, S. Wadi, D. Pandalai, and H. Spaanenberg. A benchmark suite for evaluating configurable computing systems– status, reflections, and future directions. In *Proceedings of the 2000 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA'00)*, pages 126–134, February 2000.

[2] Prashant R. Chandra and Seow Yin Lim. Framework for benchmarking network processors, September 2000. http://www.npforum.org/techinfo/NPFFramework.pdf.