TOWARDS BENCHMARKING ENERGY EFFICIENCY OF RECONFIGURABLE ARCHITECTURES

Tobias Becker, Peter Jamieson, Wayne Luk
Department of Computing
Imperial College London
{tbecker,pjamieso,wl}@doc.ic.ac.uk

Peter Y. K. Cheung
Department of EEE
Imperial College London
p.cheung@imperial.ac.uk

Tero Rissa
Nokia Devices R&D
Finland
tero.rissa@nokia.com

ABSTRACT

Energy research in reconfigurable architectures often involves legacy benchmarks such as the MCNC benchmarks. These benchmarks, however, are not well-suited for assessing energy consumption of reconfigurable technology, since they lack realistic input stimuli. This paper reviews and categorises a range of computation system benchmarks, and shows that there are no comprehensive benchmarks targeting reconfigurable architectures that would stimulate energy or power research. We review existing energy research in the field which involves microbenchmarks, in-house designs, or legacy benchmark suites used to evaluate power optimisations.

1. INTRODUCTION

Reconfigurable architectures are an emerging technology that offers promising advantages such as increased flexibility and reduced time-to-market for consumer applications. However, to meet the strong energy budget constraints of hand-held mobile devices, the energy efficiency of current reconfigurable devices needs to be improved. We therefore advocate the development of a benchmark suite that stimulates the exploration of reconfigurable architectures and allows objective evaluation of advances in energy efficiency.

In this paper, we first review existing benchmark suites for computation systems in general. We categorise these benchmarks based on the computation systems they target, and we further differentiate benchmark suites based on other characteristics.

We then analyse the benchmark suites that have been used to benchmark power and energy on reconfigurable devices. We show that even though there are some benchmark suites that have been used to measure power, these are not sufficient to stimulate advances in reconfigurable devices, since they are limited to a particular architecture and do not represent realistic application scenarios. This complicates the comparison of achievements in low-power architectures and design methods, and hinders general advances in the field.

The remainder of this paper is organised as follows. Section 2 defines a benchmark and covers some of the common challenges associated with creating a good benchmark suite. Section 3 describes the categories and characteristics for benchmark suites, and a number of existing benchmark suites. Section 4 studies benchmark suites with special focus on power and energy. Section 5 gives an overview of reconfigurable architectures and low-power research on such devices. Finally, section 6 concludes the paper.

2. BACKGROUND

Benchmarking, for many types of systems, is the process by which a system under test (SUT) takes in predefined inputs, known as a benchmark, and performs an action on this benchmark producing measurable outputs. This is, usually, repeated for each benchmark in a collection of benchmarks called a benchmark suite. Common actions for SUT include executing an algorithm for a given set of inputs or serving requests from a client.

In many cases, the goal of benchmarking systems is to compare them by their recorded measurements when executing the benchmarks. For instance, a common benchmarking experiment should allow us to compare two processors, based on how long it takes them to execute a series of algorithms. Another example concerns Computer Aided Design (CAD) and how two different optimisation algorithms with a given data set transform a circuit, such that the silicon area required to implement that circuit is minimised.

Even though existing benchmark suites vary widely in their characteristics, their development faces common challenges which include:

- Creating a fair benchmark suite that is representative of actions a system will likely perform

- Preventing system or tool optimisations for a specific benchmark, while still encouraging innovation

- Having reportable measurements that reflect the performance of the system, and are useful in comparing systems

- Collecting realistic benchmarks and arranging open access to them

The first two challenges are of particular importance, because a failure to meet them can lead to useless or deceptive

optimisations. With a good benchmark, however, optimising the system for higher benchmark scores also increases its usefulness in real application scenarios. For academic research, open or at least inexpensive access is an important aspect to increase the popularity and traction of a benchmark suite.

3. EXISTING COMPUTATION SYSTEM BENCHMARKS

To provide a feel for benchmarking in general, this section surveys many benchmark suites for computation systems and their tools. Our survey includes a wide range of benchmarks created both by companies to accredit systems and by academia to facilitate research comparison. For this reason we first categorise benchmark suites based on the types of computations systems they target. The broad range of computation system benchmarks explored in this survey is used to show the strengths of existing benchmark suites that do not target reconfigurable architectures that might then be applied to a new low energy benchmark suite targeting these technologies.
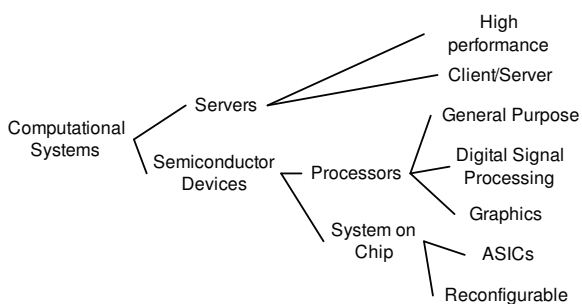
Fig. 1. Benchmark categories based on the systems that they target

Figure 1 shows hierarchical benchmark categories based on the benchmark suite's intended target systems. There are cases in which a benchmark does not fit in one target category, since these benchmarks are intended to compare more than one type of system, and in these cases we will use a category branch that contains all the targets.

In addition to categorisation, we further distinguish benchmarks in terms of workload type where this characterisation describes how the benchmarks will be executed on the SUT with respect to time. The workload options are throughput experiments in which the workload is to be executed without interruption assuming that all input data is available at the beginning of the action, transaction based in which the workload is controlled by requests made by clients or other entities, or intermittent in which the workload is predefined to occur over a range of time and the inputs are provided accordingly. In a few cases, the benchmark suite uses more than one type of workload.

Table 1. The studied benchmarks

Benchmark Name	Category	Workload type
SPEC MPI2007 [4]	High Performance	throughput
SPEC power_ssj2008 [5]	Client/Server	throughput, intermittent
EEMBC [1]	Processors	throughput
SPEC CPU2006 [13]	General Purpose	throughput
BDTi Communications [6]	DSP, FPGAs	throughput
BDTi Video Encoder and Decoder [7]	Semiconductor	throughput
BDTi Solution [7]	Semiconductor	throughput
SCU-RTL [3]	System-on-Chip	throughput
Texas97 [2]	System-on-Chip	throughput
MCNC [20]	System-on-Chip	throughput
RAW [9]	Reconfigurable	throughput
PREP [16]	Reconfigurable	throughput

Table 1 contains the categorisation of a sample of the benchmarks surveyed. In the first column, we show the name of the benchmark. The next column lists the category of the benchmark suite based on the categories shown in Figure 1. Additionally, we arrange the benchmarks in each category such that the top being the most recent and the bottom being the oldest. Column 3 describes the type of workload each benchmark uses.

4. BENCHMARKING FOR ENERGY CONSUMPTION

For many computation systems, improving area and performance has been the focus with a secondary concern on energy consumption of the system. With increasing computation demands in consumer mobile devices and limited advances in battery capacity, power and energy have become important criteria to design for. Energy or average power are relevant in the context of battery capacity while peak power has to be considered for thermal aspects. In the scope of this paper, we do not strictly distinguish between low-power and low-energy because these aspects are closely related.

Table 2 shows benchmarks that have been used to measure energy or power including some additional characterisations of these suites based on power. The first and second column show the name and category of the benchmark suite. Column three indicates whether the suite includes input stimuli, and column four indicates whether the benchmark was originally created for energy based benchmarking.

Input stimuli are necessary for measuring power or energy since the system needs to perform its required set of actions that consume power. Note in Table 2 that the commercial benchmark suites include input stimuli, and the four academic benchmark suites targeting System on Chips (SoCs)

Table 2. Benchmarks measuring either power or energy

Benchmark Name	Category	Includes Input Stimuli	Originally For Energy Benchmarking
SPEC power_ssj2008	Client/Server	Yes	Yes
JouleSort	Client/Server	Yes	Yes
EEMBC	Processors	Yes	No
BDTi DSP Kernel	DSP	Yes	No
BDTi Video Kernel	Semi. Devices	Yes	No
BDTi Video Encoder and Decoder	Semi. Devices	Yes	No
SCU-RTL	SoC	No	No
Texas97	SoC	No	No
MCNC	SoC	No	No

do not include input stimuli. In this case, realistic stimuli have to be generated by the benchmark user or statistical analysis of switching activity has to be used.

Another relevant part of energy benchmarking is the type of workload. This is especially true depending on where the SUT will be used. For example, many devices both portable and non-portable will be powered down into an idle mode (among other power modes). An SUT in idle mode will have a very different energy profile compared to the SUT in full throughput mode. The benchmark suites listed in Table 2 all have throughput type of workloads except the SPEC power_ssj2008. SPEC power_ssj2008 targets Client/Server systems and its workload is based on clients making requests to the SUT at various times at varying throughput demands. This benchmark suite's run methodology states that the benchmark will measure energy under idle, maximum throughput, and various throughput demands to simulate the nature of a server.

5. POWER AND RECONFIGURABLE ARCHITECTURES

There is a range of different reconfigurable technologies available today and in this section we briefly review fine-grain, course-grain, multi-core, and parametric ASICs. We review relevant work in reconfigurable power research to illustrate how these architectures are evaluated.

Fine-grain configurable FPGAs and CPLDs are the most common form of reconfigurable devices that usually consist of a grid of programmable logic and routing resources. These are efficient for data-flow oriented bit-level algorithms. The fabric often includes dedicated blocks such as memories, processors and DSPs that allow more efficient implementation of algorithms requiring these functions. FPGAs can significantly shorten and simplify the design process over traditional ASIC design and most FPGAs can be field-upgraded (in some cases run-time reconfigured within a millisecond timeframe).

Coarse-grain configurable devices consist of byte or word-length programmable routing architectures with more complex functional units such as memories, ALUs and Multiplier Accumulaters (MACs). These devices usually target high-performance streaming applications such as video or signal-processing with the potential for being more power efficient than FPGAs for these applications. There are different programming paradigms for these coarse-grain devices, but many of them are based on data-flow oriented descriptions.

Another approach is to add reconfigurability to ASICs [14]. A set of functions is analysed for common circuit components, and instead of switching between entire functions they are reconfigured by programmable connections and parameters of these components. This technology is arguably the most power efficient. On the other hand it suffers the drawbacks of the ASIC design process and is only reconfigurable for a pre-defined set of functions.

The power efficiency and degree of reconfigurability also depend on memory technology. Anti-fuse devices are non-volatile and power and area efficient, but they can only be programmed once. SRAM-based devices offer fast dynamic reconfiguration with the downside of area and power overheads. Devices based on Flash memory are non-volatile but also offer reprogrammability. However, configuration speeds are much lower than for SRAM-based devices.

There is extensive research on general low-power techniques for FPGAs. An early work is by George *et. al.* [12], and they create a low energy FPGA through architecture and low-level circuit design. To compare their FPGAs to Xilinx and Altera devices, they implement three simple circuits and evaluate the power consumption using Synopsis' Powermill. The three circuits consist of single flip-flop driving 9 routing segments, a 1K array of 16 bit counters, and a toggle circuit.

Instead of using simple benchmarks, Shang *et. al.* [17] show the dynamic power consumption of a Xilinx Virtex-II FPGA [19] using one Xilinx internal benchmark that represents a large industrial circuit. This internal benchmark includes input stimuli which they use to calculate the switching activity of a real design.

The above examples use simple benchmarks or no benchmark at all. The MCNC benchmark suite (see Table 1) provides a range of simple test circuits and is often used in power-aware research. Poon *et. al.* [15] use the MCNC benchmark and add power models to a common FPGA architecture exploration tool, VPR [10]. They use the MCNC benchmark suite to find a transition density signal model to estimate the activity within each logic cell of an FPGA. Anderson *et. al.* [8] also use the MCNC benchmarks in their work to estimate power consumption in FPGAs. Gayasen *et. al.* propose a scheme with two programmable supply voltages where the higher voltage is used for critical path logic and the lower voltage for non-critical parts [11].

Using MCNC circuits, they achieve an average power saving of 61 %. Even more recently, Tinmaung *et. al.* [18] optimise for power on FPGAs during logic synthesis and use the MCNC benchmarks to perform measurements of their optimisations.

In this small sample of FPGA power research, we see that the MCNC benchmark suite is a popular choice to test and measure optimisations. However, one could argue that a collection of state machines and combinatorial logic is not representative for many applications. A more severe problem is that MCNC benchmarks are specified in RTL. Even though this allows easy synthesis of the benchmark, it restricts possible target architectures. MCNC cannot capture the advantage of efficient dedicated blocks such as embedded memories or DSPs and can also not be used on more coarse grain architectures. Another drawback is the lack of input stimuli in the benchmark. Results are based on user generated stimuli or estimated switching activities. Furthermore, power is only measured in active mode for instance when the system is running at maximum throughput. This does not capture the influence of potential sleep or standby modes which is a strong concern for mobile applications. OpenCores can provide more realistic test circuits but otherwise the same limitations apply.

6. CONCLUSIONS

This paper reviews existing benchmark suites used to measure and compare computation systems. We have defined a set of categories based on what systems benchmark suites target, and then categorised existing benchmark suites based on this classification.

Our review illustrates that current benchmarks are not adequate in objectively benchmarking power and energy in reconfigurable devices. In order to be effective, a benchmark suite should be based on a realistic set of test scenarios that are representative for the target application. The benchmarks should also be specified at a level of abstraction that allows the exploration of different implementations and architectures. Finally, realistic workloads are needed to evaluate energy efficiency not only for maximum throughput but also in different low-power or idle modes.

7. REFERENCES

[1] Embedded Microprocessor Benchmark Consortium (EEMBC). http://www.eembc.org..

[2] http://www-cad.eecs.berkeley.edu/Respep/Research/vis/texas-97/, 1997.

[3] http://www.engr.scu.edu/mourad/benchmark/RTL-Bench.html, 1998.

[4] SPEC MPI2007. http://www.spec.org/mpi2007/, 2007.

[5] SPECpower_ssj2008. http://www.spec.org/power_ssj2008/, 2007.

[6] BDTI Communications Benchmark. http://www.bdti.com/products/services_comm_benchmark.html, 2008.

[7] BDTI Video Encoder and Decoder Benchmarks. http://www.bdti.com/products/services_video_benchmark.html, 2008.

[8] J. Anderson and F. Najm. Power estimation techniques for FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(10):1015–1027, October 2004.

[9] J. Babb, M. Frank, E. Waingold, R. Barua, M. Taylor, J. Kim, S. Devabhaktuni, and A. Agarwal. The RAW Benchmark Suite: Computation Structures for General Purpose Computing. In *IEEE Symposium on FPGAs for Custom Computing Machines*, pages 134–143, 1997.

[10] V. Betz and J. Rose. Directional Bias and Non-Uniformity in FPGA Global Routing Architectures. In *14th IEEE/ACM Int'l Conference on CAD*, pages 652–659, 1996.

[11] A. Gayasen, K. Lee, N. Vijaykrishnan, M. Kandemir, M. Irwin, and T. Tuan. A dual-Vdd low power FPGA architecture. In *Field Programmable Logic and Application*, pages 145–157, 2004.

[12] V. George, H. Zhang, and J. Rabaey. The design of a low energy FPGA. In *ISLPED '99: Proceedings of the 1999 international symposium on Low power electronics and design*, pages 188–193, 1999.

[13] J. L. Henning. SPEC CPU2006 Benchmark Descriptions. *ACM SIGARCH Computer Architecture News*, 34(4):118–121, September 2007.

[14] T. Makimoto and Y. Sakai. Evolution of low power electronics and its future applications. In *International symposium on Low power electronics and design*, pages 2–5, 2003.

[15] K. Poon, A. Yan, and S. Wilton. A Flexible Power Model for FPGAs. In *Field-Programmable Logic and Applications*, pages 312–321, 2002.

[16] Programmable Electronics Performance Corporation. PLD Benchmark Suite#1, V1.2. 504 Nino Ave. Los Gatos, CA 95032, 1993.

[17] L. Shang, A. S. Kaviani, and K. Bathala. Dynamic power consumption in Virtex™-II FPGA family. In *FPGA '02: Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, pages 157–164, 2002.

[18] K. O. Tinmaung, D. Howland, and R. Tessier. Power-aware FPGA logic synthesis using binary decision diagrams. In *FPGA '07: Proceedings of the 2007 ACM/SIGDA 15th international symposium on Field programmable gate arrays*, pages 148–155, 2007.

[19] Xilinx. *Virtex-II Pro Platform FPGAs: Functional Description*, Oct 2003.

[20] S. Yang. Logic Synthesis and Optimization Benchmarks, Version 3.0. Tech. Report. Microelectronics Centre of North Carolina. P.O. Box 12889, Research Triangle Park, NC 27709 USA, 1991.