DSP QUANT: DESIGN, VALIDATION, AND APPLICATIONS OF DSP HARD REAL-TIME BENCHMARK

Chunho Lee, Darko Kirovski, Inki Hong and Miodrag Potkonjak

UCLA Computer Science Department, Los Angeles, CA 90095-1596, USA

ABSTRACT

Although the undeniable importance of high quality, efficient and effective DSP synthesis benchmark has been firmly and widely established, until now the emphasis of benchmarking has been restricted on assembling individual examples. In this paper we introduce the "ideal candidate benchmark methodology" which poses the development of the benchmark as well as defines a statistical and optimization problem. We first outline the goals and requirements relevant for the benchmark development. After discussing the computational complexity of the benchmark selection problem, we present a simulated annealing-based algorithm for solving this computationally intractable optimization task. Using this approach from 150 examples we select 12 examples for the new DSP Quant benchmark for DSP hard Real-Time applications. The DSP benchmark is statistically validated, and its application to the analysis and development of system-level synthesis algorithms is demonstrated.

1. MOTIVATION

Benchmarking is a widely recognized and extensively used approach to performance evaluation. Since the early beginning of computer science and engineering, benchmarking has been playing a wide variety of important roles, greatly influencing major hardware and software concepts. A number DSP benchmarks are presented in [8] [9], but their quality and performances have never been reported. Recently, benchmarking attracted an exceptionally high attention in both research and industrial CAD communities. Popular benchmarks are available for sequential test generation, logic synthesis, physical design, and circuit simulation [1] [2].

We have developed a new quantitave methodology for the design and statistical validation of benchmarks. The approach and optimization algorithms are used for the development of the new DSP Quant benchmark for DSP hard real-time application specific systems. The benchmark is statistically validated and methodology of benchmark used for synthesis tool development is demonstrated.

The rest of the paper is organized in the following order. The target applications are discussed in Section 2. The criteria for the benchmark development methodology is described in Section 3. Section 4 formulates the benchmark selection process as an optimization problem and Section 5 provides the synthesis algorithm. In Section 6 we discuss the validation issue and provide experimental results. Finally, Section 7 draws conclusions.

2. TARGET APPLICATIONS

We envision the following applications of quantitave statistically validated benchmarks.

- Comparison of synthesis systems and tool selection: The goal is not only to establish proper ranking among the available synthesis tools, but also, more importantly, to identify which synthesis system will most likely perform well for a given mix of customer applications.
- Fine-tuning of synthesis software: The information from benchmark tests can be used for the fine-tuning purposes during the development of synthesis software. Benchmarks provide a good guidance how to tune parameters in the algorithms of synthesis and compilation tools to the characteristics of typical real life designs. This practice, however, may cause an undesirable side effect called "over-tuning".
- Software validation and verification: Benchmarks are often invaluable aid in "software system maintenance and debugging". When we have a benchmark which

we can always use to test a program when something is changed, we can be more assured to significantly higher degree that no new bugs are introduced by the fix of the bugs.

- Improvement of synthesis tools: Benchmark should be capable to pinpoint a particular weakness of a particular tool. For example, it should be able to characterize the type of examples for which a given tool does not perform as well as other tested tools.
- Research guidance: A well designed benchmark should help researchers to identify what the key high-impact research areas are and what design goals to optimize. For example, this can be done by comparing the current results of the available tools to the minimum-bounds of what can be achieved or by identifying where the key advantages of better performing tools are coming from.

3. BUILDING HARD REAL-TIME DSP SYNTHESIS BENCHMARK

Using the goals established in the previous section, we develop the following criteria which guide the benchmark development methodology.

- Relevance criteria: The benchmark should be based on real life applications representing their full diversity and complexity.
- Compactness criteria: Benchmark should contain as few designs as possible. The implementation of design specifications and their synthesis are often time consuming.
- Comprehensiveness criteria: Although compactness, as just stated, is important, it is also important to cover all types of common designs. The good benchmark strikes balance between its size and comprehensiveness.
- Resolution criteria: Benchmark should enable fair comparison between competing tools and approaches with good resolution. In other words, we do not want all the examples too easy nor too difficult so that we are able to clearly differentiate the quality of the various systems being compared.
- User feedback criteria: We want to provide exact characterization of the types of examples on which particular tool performs significantly better or worse than other tools.

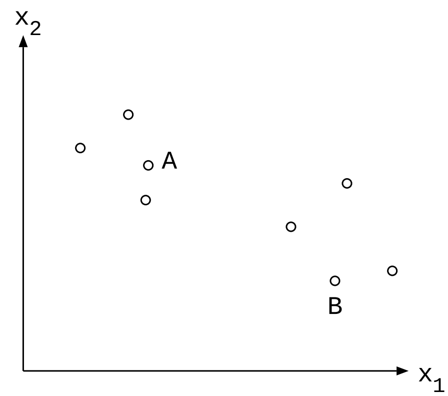

Figure 1. A design example space and candidate benchmarks (indicated by A and B)

4. IDEAL CANDIDATES BENCHMARKING PROBLEM FORMULATION

The new Ideal Candidates Benchmarking(ICB) problem can be stated at the conceptual level in the following way.

"A sufficiently large set of N examples is given and each example is numerically characterized by the set of relevant properties. The task is to select k examples which according to some quantitative criteria are the best descriptor of all examples".

The intuition behind the formulation is clear: We want to select a small subset of examples which ideally represent all available examples, so that later on we can extract all the information which can be obtained using all the examples by considering only ideal candidates. It is assumed that two examples are similar if there is a little (or no) discrepancy between them in all important parameters relevant to the targeted synthesis process.

As an illustration, an example is shown in Figure 1. A set of design examples is distributed in a design example space. x_1 and x_2 can be any design properties. There are two discernible clusters in the design example space. There are such a pair of design examples, indicated by A and B, that better represent the two clusters than any other pair of two examples.

Other underlying assumptions about the available examples are the following: (i) The available pool of examples is reasonably complete in the sense that any new example of the interest is similar to at least one of the examples in its diversity and its distribution in the design space. This assumption can be later on tested in a statistically sound way by analyzing the distribution of the properties for the available examples. (ii) New (missing) examples can be found and added if there are any so that the available pool of examples can be completed.

We now define the ICB problem using more formal Garey-Johnson format [4].

ICB Problem:

Instance: A sufficiently large set of N examples is given and each example x_j is numerically characterized by the set of m relevant properties, x_{jl}, $l = 1, ..., m$ and a constant M.

Question: Are there k examples y_i, $i = 1, ..., k$ such that $\sum_{i=1}^{k} D(y_i) \leq M$, where $D(y_p) = \min_{j=1,...,N where x_j \neq y_i, \forall i=1,...,k} Dist(y_p, x_j)$. $Dist(y_i, x_j)$ can be any distance measure, for example, geometric $\sqrt{\sum_{l=1}^{m}(y_{il} - x_{jl})^2}$, or Manhattan $\sum_{l=1}^{m} |y_{il} - x_{jl}|$.

We proved that the ICB problem is NP-complete by using polynomial "local replacement" [4] transformation from a special instance of the MIN-MAX MULTICENTER problem where the selected points are from the set of the available points.

5. SYNTHESIS OF ICB PROBLEM

```
Benchmark Selection()
  Extract all relevant properties from all
  available designs;
  Determine range for each property;
  Normalize the values of each property against
  the largest in the set;
  Run the simulated annealing based algorithm
  to choose a subset for the benchmark;
```

Figure 2. A pseudo-code of the algorithm

The overall procedure for the quantitative development of benchmarks which includes a general combinatorial optimization technique known as simulated annealing [7], is given by the pseudo-code in Figure 2. The actual implementation details of the simulated annealing algorithm are presented for each of the following areas; the cost function, the neighbor solution generation, the temperature update function, the equilibrium criterion and the frozen criterion. Firstly, various distance measures including geometric and Manhattan have been used as the cost function. All of them produced similar results. Secondly, the neighbor solution is generated by the interchange between a randomly chosen design example in the current solution and another randomly chosen design example not in the current solution. Thirdly, the temperature is updated by the function $T_{new} = \alpha(T_{old}) * T_{old}$. For the temperature $T > 200.0$, α is chosen to be 0.1 so that in high temperature region where every new state has very high chance of acceptance, the temperature reduction occurs very rapidly. For $1.0 < T \leq 200.0$, α is set to 0.95 so that the optimization process explores this promising region more slowly. For $T \leq 1.0$, α is set to 0.8 so that T is relatively quickly reduced to converge to a local minimum. The initial temperature is set to 4,000,000. Fourthly, the equilibrium criterion is specified by the number of iterations of the inner loop. The number of iterations of the inner loop is set to 30. Lastly, the frozen criterion is given by the temperature. If the temperature falls below 0.1, the simulated annealing algorithm stops.

The relevant properties we used are the following (for detailed definition of properties see [6]); (i) Number of Operations (ii) Iteration Bound (iii) Variance of the number of operations for a given task (iv) Average Parallelism (v) Maximal Parallelism (vi) Percentage of Operations which can be pipelined; (vii) Average Temporal Locality (viii) Maximal Temporal Locality.

1. NEC digital to analog converter
2. Motorola's 126-tap FIR filter
3. IMEC modem
4. 7th order WDF filter
5. 8 point 1 dimension decimation-in-time DCT
6. 5th order distillation plant controller
7. 2nd order Volterra filter
8. (31, 15)BCH code
9. 2D 8 point VHS transform
10. Hilbert transform
11. 2-D 10-th order IIR filter
12. 9-order rank-order filter.

Figure 3. The benchmark of DSP Quant Version 0.1

In order to properly characterize the typical media design examples, we assembled 150 DSP, video, image, graphics, telecommunication, and information theory application examples. The assembled design set includes three versions of the low pass 5th order elliptical wave digital filter, 7th and 9th order wave digital filters, Winograd Convolution, several different forms (direct, cascade, parallel, continued fraction and Gray-Markel ladder) 8th order IIR bandpass Avenhaus filters, several different Winograd transformations, several 1D and 2D DCT algorithms, VHS transform, NTSC coder, G2 blend surface calculation using quintic polynomials, LMS adaptive filter, Hilbert Transform, 2nd order Volterra filter, 3rd order truncated Volterra filter, 28th, 50th, 60th, 100th, 126th and 133rd order FIR filters with a variety of transfer function and applied windowing techniques, PCM DA converter, image histogram, two GE 5th order linear controllers, steam controller, power controller, 3 VSTOL airplane controllers, distillation plant controller, Goppa, Reed-Muller, Fire, Hamming and BCH code. The benchmark of the DSP Quant preliminary version 0.1 which consists of 12 examples is provided in Figure 3.

6. VALIDATION AND APPLICATIONS

We validated the proposed approach and the simulated annealing algorithm for the ideal candidates benchmark using both a statistical method and a direct application in system level synthesis.

The validation of the DSP Quant benchmark has been conducted using a statistical validation technique, resubstitutions [3]. A number of randomly chosen design examples are eliminated from the pool of design examples from which the original benchmark set is selected. A new benchmark set is selected. By comparing the new benchmark set with the original one by the "weighted bipartite matching" technique, we can measure the validity of the original and the new ones. We use *the Hungarian Method* [5] to solve the problem.

Even more important than the statistical validation is the experimental demonstration of the effectiveness of the the DSP Quant benchmark. We recently developed a scheduling and assignment program for power minimization on programmable heterogeneous platforms. The program uses a set of 9 parameters. The performance of the program varies over factor of almost two orders of magnitude, depending on the selected values for the parameters. The values of the parameters are iteratively set by using Gauss-Jordan iterative procedure through the examination of the performances on a set of learning examples. We were able to tune the parameters of our scheduling and assignment program to the best performing values using the selected 12 examples. The same level of performance was achievable only when we randomly selected between 41 and 83 examples (average 60), clearly indicating the expression power of the selected benchmark examples.

7. CONCLUSION

Based on the set of goals and requirements for the effective and efficient benchmark development, we have formulated the ideal candidates benchmark approach. The computational complexity of the associated optimization problem has been established, and the simulated annealing-based algorithm was successfully used for solving the NP-complete problem. The comprehensive approach for the benchmark validation was described and the effectiveness of the benchmark was demonstrated by the application of the benchmark to system-level synthesis algorithms and software.

REFERENCES

[1] F. Brglez, D. Bryan, K. Kozminski, "Combinational profiles of sequential benchmark circuits", International Symposium on Circuits and Systems, pp. 1929-1934, 1989.

[2] F. Brglez: "ACM/SIGDA Design Automation Benchmarks: Catalyst or Anathema?", IEEE Design & Test of Computers, pp. 87-91, September 1993.

[3] B. Efron, R.J. Tibshirani, "An introduction to the bootstrap", Chapman & Hall, New York, NY, 1993.

[4] M.R. Garey, D.S. Johnson: "Computers and Intractability", W.H. Freeman, New York, NY, 1979.

[5] H. W. Kuhn, "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, Vol. 2, pp. 83-97, 1955.

[6] L. Guerra, M. Potkonjak, J. Rabaey, "System-level design guidance using algorithm properties", VLSI Signal Processing VII, pp. 73-82, 1994.

[7] S. Kirkpatrick, C. Gelatt, M. Vecchi, "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671-680, 1983.

[8] D. Shear: "EDN's DSP Benchmarks", EDN, pp. 126-148, Sept. 29, 1988.

[9] V. Zivojnovic, et al, "DSPstone: A DSP-Oriented Benchmarking Methodology", International Conference on Signal Processing and Technology, pp. 715-720, 1994.