A Framework for Design Space Exploration of Parameterized VLSI Systems

Giuseppe Ascia Vincenzo Catania Maurizio Palesi

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Università di Catania
V.le Andrea Doria, 6 – 95125 Catania - Italy
{gascia,vcatania,mpalesi}@diit.unict.it

Abstract

The paper presents two new approaches to multi-objective design space exploration for parametric VLSI systems. Both considerably reduce the number of simulations needed to determine the Pareto-optimal set as compared with an exhaustive approach. The first uses sensitivity analysis while the second uses evolutionary computing techniques. Application to a highly parametric system-on-a-chip for digital camera applications shows the validity of the methodologies presented in terms of both accuracy of results and efficiency, measured as the number of simulations needed to determine the power/execution-time trade-off front.

1 Introduction

Current technology makes it possible to integrate many more transistors than a VLSI system designer can use on a single silicon die. If design methodologies remain unchanged, the productivity gap will become increasingly evident [2]. A comparison can be made with what happened with software from the 80s onwards — the sudden abundance of memory shifted the design paradigm from optimisation in size towards methodologies that would allow for portability and reuse (first in the form of parametric subroutines and then in the form of parametric objects). In turn, a new approach is being introduced in the hardware sector — IP-based design in which a highly parametric pre-designed system-on-a-chip (SOC) is configured according to the application it will have to execute.

This new approach called *configure-and-execute* [12] is based on the presence of highly parametric IPs (Intellectual Properties) representing the basic components of a SOC. Once the architecture of a system has been designed, that is, it has been decided which IPs to use, it is necessary to find the optimal configuration for them according to the specific application (or set of applications) that have to be executed. The values chosen for these parameters (bus sizes, coding techniques, cache parameters, arbitration schemes, etc.) are those that optimise a function which almost always depends on three main variables: area, power and performance.

The greatest problems in this area regard exploration of the range of possible system configurations in search of the optimal configuration for a given application. There are, in fact, a number of parameters involved (bus sizes, cache configurations, software algorithms, etc.), each of which has a great impact on design constraints such as area, power and performance. An exhaustive analysis of all possible configurations is thus computationally unfeasible.

Research in the field of parameterized system design has led to the definition of various approaches to explore the range of configurations. In [6] sensitivity analysis was used to search for the configuration that minimises the power-delay product for a cache memory. In [3] mono-objective genetic algorithms (GAs) were used to search for optimal configurations in terms of area, power and average access time for a memory hierarchy. In [7] a system comprising a CPU, caches and main memory and the interfaces between these cores was analysed to show the power-performance trade-off for various technologies. Of course, any technique used to explore a range of configurations requires tools to evaluate the configurations and thus system-level simulation and estimation techniques. These tools have to be capable of performing system-level simulations in as short a time as possible as well as estimating variables that are typical of a lower level of abstraction (e.g. clock cycles, power consumption) with an adequate level of accuracy. In [8] the authors used power estimation data obtained from the gate-level for a cores representative input stimuli data, and propagated this data to a higher (object-oriented) system-level model, which is parameterizable and executable. They achieved simulation speedups of over 1000 with accuracies suitable for making reliable power-related system-level design decisions. In [9] the same authors describe a method

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 IEEE

for speeding up the evaluation further, through the use of instruction traces and trace simulators for every core, not just the microprocessor core.

The aim of this paper is to present a general methodology to search for the Pareto-optimal set of configurations of a parameterized system that optimise the system in relation to various objectives. The methodology uses multi-objective optimisation techniques based on genetic algorithms. The results obtained in a case study (a highly parametric SOC for digital camera applications) show the efficiency of the approach in terms of both accuracy and the number of simulations required for the exploration.

The paper is structured as follows. In Section 2 the problem will be formally stated and a survey of multi-objective optimisation techniques based on genetic algorithms will be given. In Section 3 two new approaches to design space exploration will be presented and then applied to determine the power/performance trade-off for a highly parametric system described in Section 4 together with the results obtained. Finally, Section 5 provides our conclusions and indications as to future developments.

2 Statement of the Problem

Exploration of a range of configurations for a parameterized system involves multiple measures of performance, or objectives (in a VLSI system, for example, area, power and delay...) which should be optimised simultaneously. Often optimal performance according to one objective, if such an optimum exists, implies unacceptably low performance in one or more of the other objective dimensions, creating a need for a compromise to be reached.

Exploration of a range of configurations for a parameterized system can be defined as a set of techniques and strategies to be used to determine *mutually non-dominated* configurations (the concept of dominance will be explained below). The solution to these problems falls into the multi-objective optimisation strategy class. Multiobjective optimisation (also called multicriteria optimisation, multiperformance or vector evaluation) can be defined as the problem of finding [4]: "a vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimise" means finding a solution which would give values for all the objective functions such as to be acceptable to the designer."

In formal terms we can define the problem in this way: find the vector $\overline{c}^* = [p_1^*, p_2^*, \ldots, p_P^*]^T$ which will satisfy the m inequality constraints:

$$g_i(\overline{c}) \geq 0 \; i = 1, 2, \ldots, m \tag{1}$$

the n equality constraints

$$h_i(\overline{c}) = 0 \; i = 1, 2, \ldots, n \tag{2}$$

and optimizes the vector function

$$\overline{f}(\overline{c}) = [f_1(\overline{c}), f_2(\overline{c}), \ldots, f_N(\overline{c})]^T \tag{3}$$

where $\overline{c} = [p_1, p_2, \ldots, p_P]^T$ is the vector of decision variables.

That is, it is necessary to extract from the set $\mathcal{F}$ vectors $\overline{c}^*$ that will satisfy 1 and 2 the vectors that optimise all the components of the objective function vector $\overline{f}(\overline{c})$.

Multiobjective optimisation dates back to Vilfredo Pareto's treatise on political economy in [10]. We say that a point $\overline{c}^* \in \mathcal{F}$ is Pareto optimal if for every $\overline{c} \in \mathcal{F}$ either,

$$\bigwedge_{i \in I} (f_i(\overline{c}) = f_i(\overline{c}^*))$$

or, there is at least one $i \in I$ such that

$$f_i(\overline{c}) > f_i(\overline{c}^*)$$

This definition states that $\overline{c}^*$ is Pareto optimal if there exist no vectors $\overline{c} \in \mathcal{F}$ that decrease the value of any component of the cost function (assuming that the objective function is a cost function to be minimised) without increasing the value of another component of the cost function. Unfortunately, the Pareto optimum is not a single one but a set of solutions called *non-inferior* or *non-dominated* solutions.

3 Design Space Exploration

The main aim in defining a methodology for design space exploration (DSE) of a parametric system is to generate the Pareto-optimal set of configurations that optimise towards several objectives. Evaluation of a generic configuration to determine the parameter values towards which optimisation should aim requires configuration and simulation of the system. Simulation of a complex system is generally a computationally onerous operation in terms of CPU time. So a methodology based on an exhaustive search for the Pareto-optimal set in unfeasible since the space of configurations is equal to the product of the cardinalities of the sets of values each parameter takes.

The use of heuristic techniques can help to reduce the space of configurations that have to be analysed by identifying and discarding any Pareto-dominated [6] configurations. Another approach presented in [3] proposes the use of genetic algorithms as an efficient technique for DSE.

Both techniques reduce the problem of multi-objective optimisation to one of scalar optimisation by aggregation of the objective functions [13, 11].

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 IEEE

The main disadvantage to aggregation functions is that they do not generate proper Pareto-optimal solutions in the presence of non-convex search spaces, which is a serious drawback in most real-world applications. These problems are solved by Pareto-based approaches that select Pareto non-dominated individuals from the rest of the population. These individuals are then assigned the hightest rank and eliminated from further contention. Another set of Pareto non-dominated individuals are determined from the remaning population and are assigned the next hightest rank. The procedure is repeated until the whole population is suitably ranked.

In the following subsections, the approaches based on sensitivity analysis and GAs will be extended to conduct a proper multi-criteria analysis of the notion of Pareto optimum.

3.1 Pareto-Based Sensitivity Analysis

The sensitivity analysis (SA) presented in [6] reduces the space of possible configurations in two phases. The aim of the first phase is to identify the parameters which most influence the objective function to be optimised (sensitivity analysis phase (SAP)). For a system with P parameters, determination of the degree of sensitivity of each parameter consists of fixing $P-1$ parameters and varying one of them, determining the maximum range of variation of the objective function. One way to fix the parameters is to consider the mean value of their variation set.

The next phase, design space exploration phase (DSEP), identifies the optimal value for each parameter, from the most to the least sensitive. If $\mathcal{V}^{(i)} = \{v_1^{(i)}, v_2^{(i)}, \ldots, v_{N_i}^{(i)}\}$ is the set of values the parameter p_i can take, the number of configurations to be evaluated goes down from $\prod_{i=1}^{P} N_i$ to $\sum_{i=1}^{P} N_i$.

To understand how this approach works, let us consider the following example. Let us assume that we want to find the configuration of a cache memory in terms of size (S), block size (BS) and associativity (A) that minimises the power-delay product (PD). Let $\mathcal{S}$, $\mathcal{B}$ and $\mathcal{A}$ respectively be the set of possible values of the parameters S, B and A. The SAP is performed as follows: we fix $B = B_0$ and $A = A_0$ and S is made to vary in $\mathcal{S}$, obtaining the set $\mathcal{PD} = \{PD_1, PD_2, \ldots, PD_{|\mathcal{S}|}\}$ of values of PD. If $PD_{max} = \max \mathcal{PD}$ and $PD_{min} = \min \mathcal{PD}$, the sensitivity of the parameter S is $s_S = PD_{max} - PD_{min}$. The same procedure is repeated to determine the sensitivity of the remaining parameters s_B and s_A.

Under the hypothesis that $s_S > s_A > s_B$, the DSEP proceeds as follows. We set $A = A_0$ and $B = B_0$ and vary $S \in \mathcal{S}$, obtaining $\mathcal{PD} = \{PD_1, PD_2, \ldots, PD_{|\mathcal{S}|}\}$ values of PD. If $S_{opt} = \min \mathcal{PD}$ we fix $S = S_{opt}$ and $B = B_0$ and vary $A \in \mathcal{A}$. Proceeding as previously, A_{opt} is determined. In short, having fixed $S = S_{opt}$, $A = A_{opt}$ and varying $B \in \mathcal{B}$ we determine B_{opt}. The configuration $\langle S_{opt}, B_{opt}, A_{opt} \rangle$ will determine a PD value close to PD_{min}.

To overcome the limits of a mono-objective approach we extended the technique based on sensitivity analysis to perform multi-objective optimisation based on the notion of Pareto optimum.

The SAP was modified by defining a new metric to measure the sensitivity of a parameter. We have defined the sensitivity s_i of the i-th parameter as:

$$s_i = \max_{h,k \in \{1,2,\ldots,N_i\}} \mathrm{dist}(\overline{o}_h^{(i)}, \overline{o}_k^{(i)})$$

where dist is the Euclidean distance and $\overline{o}_j^{(i)}$ are the N_i points in the n-dimensional space obtained by fixing $P-1$ parameters and varying $p_i \in \mathcal{V}^{(i)}$.

Indicating with S_i a parameter order by decreasing degrees of sensitivity, i.e. such that $s_{S_i} > s_{S_{i+1}}$, we defined the DSEP as follows. Having fixed $p_{S_2}, \ldots, p_{S_P}$ parameters, $p_{S_1} \in \mathcal{V}^{(S_1)}$ is made to vary. From the N_{S_1} points obtained, the non-dominated configurations are extracted and accumulated in the set $\mathcal{ND}$. At the second iteration for each configuration in the set $\mathcal{ND}$, $p_{S_2} \in \mathcal{V}^{(S_2)}$ is made to vary. From the $N_{S_2} \times |\mathcal{ND}|$ obtained, the non-dominated configurations are extracted and accumulated in the set $\mathcal{ND}$. The procedure is repeated for all the parameters whose sensitivity exceeds a certain threshold. At the end of the algorithm the configurations in $\mathcal{ND}$ will represent the trade-off surface identified. The algorithm 1 gives the pseudo-code of the DSE procedure.

Algorithm 1 Pareto-based sensitivity analysis (design space exploration phase).

Require: $S_1, S_2, \ldots, S_m$ // sorted by sensitivity parameter's index
  $\mathcal{ND} = \{\overline{p}^*\}$ // initialize non-dominated set
  $i = 1$ // high sensitive parameter index
  repeat
    $\mathcal{C} = \{\}$
    for all $\overline{c} \in \mathcal{ND}$ **do**
      for all $v \in \mathcal{V}^{(S_i)}$ **do**
        $\overline{c}[S_i] = v$
        $\mathcal{C} = \mathcal{C} \bigcup \{\overline{c}\}$
      end for
    end for
    $\mathcal{ND} = \mathcal{ND} \bigcup \mathcal{C}$
    $\mathcal{ND} = \mathcal{ND} \setminus \mathrm{Dominated}(\mathcal{ND})$ // remove dominated solutions
    $i = i + 1$ // next high sensitive parameter
  until $i > P$ OR $\mathrm{Sensitivity}(S_i) < \mathrm{MINSENSITIVITY}$

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 IEEE

3.2 Pareto-based Genetic Algorithms

In this paper we have considered multiobjective optimisation techniques that use Pareto-based GAs. More specifically, we chose the *Strength Pareto EA* (SPEA) [16, 15] approach, which is very effective in sampling from along the entire Pareto-optimal front and distributing the solutions generated over the trade-off surface. The flow proposed is shown in Figure 1. An initial population of configurations is simulated to obtain an estimate of the parameters to be optimised. Together with the constraint specifications, these parameters will be used by the genetic exploration algorithm to generate the next population to be evaluated. The cycle is repeated until a stop condition is met and Pareto-optimal solutions are provided.

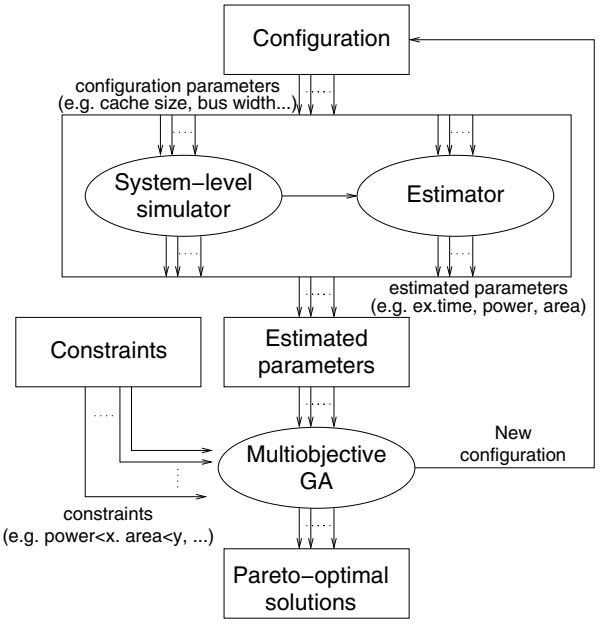

Figure 1. Design flow.

The algorithm was implemented using GAlib [14] (a C++ library of genetic algorithm components). A configuration is represented by an individual of the population whose genome defines its parameters. Each gene represents a system parameter (defined by means of an allele) that only codes values defined within the range that is admissible for the parameter involved. Impossible configurations were excluded by using the approach classified in [5] as rejection of unfeasible individuals.

4 Results

To test the methodologies proposed in Section 3 we used the architecture shown in Figure 2. It is a highly parametric SOC for digital camera applications developed under

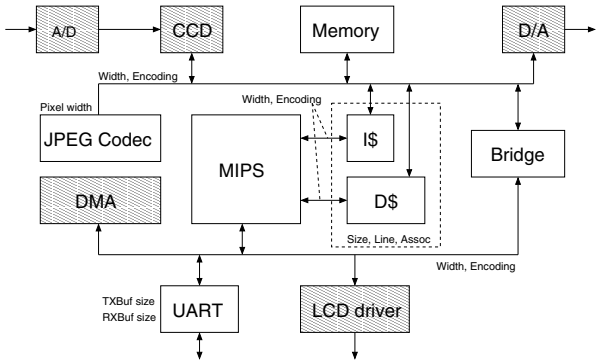

Figure 2. Reference architecture.

the Dalton Project at the University of California at Riverside [1]. The project is an open source one and comprises a parameterized simulation model of a system-on-a-chip composed of an MIPS R3000 processor core, instruction cache (I$), data cache (D$), memory, MIPS to instruction cache bus, MIPS to data cache bus, instruction/data cache to memory bus, bus bridge, peripheral bus, uart and codec.

Each core is parametric and Table 1 gives the free parameters and the set of admissible values. For each bus (data bus or address bus) it is possible to configure the number of lines and the encoding scheme to minimise the switching activity. The caches can be configured in size, line size and associativity. For the UART it is possible to define the transmission and reception buffer sizes, and for the JPEG Codec the pixel width can be varied. In all there are 26 separate parameters, giving a total of 9.7×10^{15} possible configurations.

There are two versions of the system: both a synthesisable VHDL version and a high-level model written in C++. With this model it is possible to perform rapid simulations of the system when it is executing an applications, as well as estimating the execution time and power consumption by using the estimation model described in [8].

The two methodologies were compared considering three different applications. The first, *image*, copies a bitmap from one memory region to another. The second, *key*, works on large-size matrices. The third, *matrix*, performs arithmetical operations on two 10x10 matrices of integers.

Figure 3 gives the power/execution-time trade-off front for application *key*, obtained by applying the algorithms described in the previous section. In applying the Pareto-based sensitivity analysis (PBSA) three different thresholds were used (10%, 5% and 1%). A lower threshold would improve results (as more parameters are taken into consideration) but would require a larger number of simulations. The figure also gives the results obtained by using the genetic approach (GA) after 20, 30 and 50 generations. In all

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 IEEE

Core	Parameters	Values	Number
I & D Cache	size	$128B, 256B, 512B, \ldots, 64KB$	10×2
	line	$4B, 8B, 16B, \ldots, 128B$	6×2
	associativity	$1, 2, \ldots, 16B$	5×2
I\$ & D\$ $\leftrightarrow$ CPU	dbus width	$4, 8, \ldots, 32$	4×2
	abus width	$4, 8, 16, 32$	4×2
	dbus encoding	$bin, gray, inv$	3×2
	abus encoding	$bin, gray, inv$	3×2
\$ $\leftrightarrow$ MEM	dbus width	$4, 8, \ldots, 32$	4
	abus width	$4, 8, 16, 32$	4
	dbus encoding	$bin, gray, inv$	3
	abus encoding	$bin, gray, inv$	3
Peripheral bus	dbus width	$4, 8, 16, 32$	4
	abus width	$4, 8, 16, 32$	4
	dbus encoding	$bin, gray, inv$	3
	abus encoding	$bin, gray, inv$	3
UART	TX buf size	$1, 2, 4, 8, 16$	5
	RX buf size	$1, 2, 4, 8, 16$	5
Codec	pixel width	$10, 12$	2
Global	volt vs. freq	$(1.5, 33), (2.6, 57), (3.3, 72)(4.0, 88), (5.0, 110)$	5
	Total	26	9.7×10^{15}

Table 1. Free parameters and the set of admissible values for each core.

cases the size of the internal population was set as equal to that of the external population, i.e. 50 individuals. The crossover probability was set to 0.9 and the mutation probability to 0.01. Of course, only limited weight can be given to a single run per algorithm. Nevertheless, the results were similar when the experiments were repeated with different initial populations. As can be seen, the results obtained by the GA approach after only 10 generations are better than those obtained by PBSA with a 1% threshold.

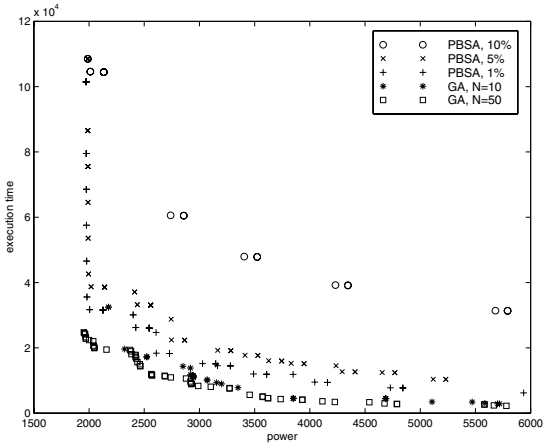

Figure 3. power/execution time trade-off front for application *key*.

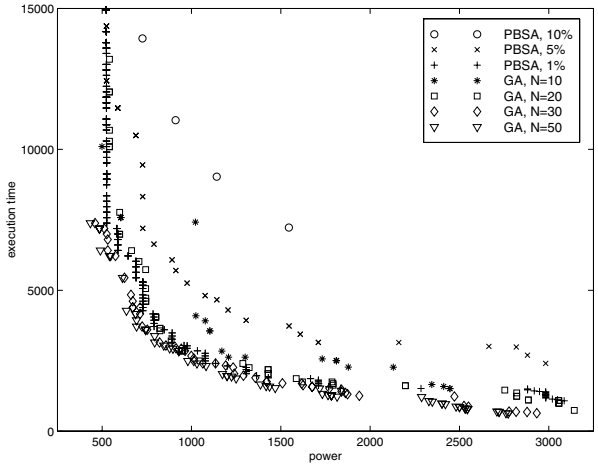

Figure 4. power/execution time trade-off front for application *image*.

Figure 4 gives the results obtained for the *image* application. Here again, the results obtained by the GA approach are better than those obtained by PBSA after only 30 generations.

A comparison between the two approaches in terms of efficiency, measured as the number of simulations run, is shown in Table 2. In all the cases considered, the GA-

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 IEEE

Application	PBSA			GA			
	10%	5%	1%	N=10	N=20	N=30	N=50
Image	142	502	9570	656	1622	2572	5082
Key	162	442	850	723	1523	2869	5016
Matrix	1024	2812	17126	700	1520	2692	4833

Table 2. Comparison between PBSA and GA in terms of efficiency measured as the number of simulations run.

based approach gave better solutions than PBSA and required fewer simulations.

5 Conclusions

This paper has presented a new genetic algorithm-based methodology for multi-objective exploration of the space of configurations of a parameterized system. The methodology has been compared with an approach based on sensitivity analysis, which we extended for multi-objective optimisation purposes.

Both approaches were applied to determine the power/performance trade-off of a highly parametric architecture implementing a SOC for digital camera applications during the execution of different applications.

The two approaches were compared in terms of both accuracy in determining the Pareto-optimal set and efficiency, measured as the number of simulation required. In all the tests performed, the power/performance trade-off generated by the GA-based approach dominated that generated by the sensitivity analysis-based approach and required on average half as many simulations.

References

[1] The UCR Dalton Project IP-Based Embedded System Design. `http://www.cs.ucr.edu/~dalton/`.

[2] International thechnology roadmap for semiconductors. Semiconductor Industry Association, 1999.

[3] G. Ascia, V. Catania, and M. Palesi. Parameterized system design based on genetic algorithms. In *9th. International Symposium on Hardware/Software Co-Design*, pages 177–182, Copenhagen, Denmark, Apr. 25–27 2001.

[4] C. A. C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, Aug. 1999.

[5] C. A. C. Coello. Treating constraints as objectives for single-objective evolutionary optimization. Technical report, Laboratorio Nacional de Informatica Avanzada, Rebsamen 80, Xalapa, Veracruz 91090, Mexico, 2000.

[6] W. Fornaciari, D. Sciuto, C. Silvano, and V. Zaccaria. A design framework to efficiently explore energy-delay tradeoffs. In *9th. International Symposium on Hardware/Software Co-Design*, pages 260–265, Copenhagen, Denmark, Apr. 25–27 2001.

[7] T. D. Givargis, J. Henkel, and F. Vahid. Interface and cache power exploration for core-based embedded system design. In *International Conference on Computer-Aided Design (ICCAD)*, pages 270–273, Nov. 1999.

[8] T. D. Givargis, F. Vahid, and J. Henkel. A hybrid approach for core-based system-level power modeling. In *Asia and South Pacific Design Automation Conference*, 2000.

[9] T. D. Givargis, F. Vahid, and J. Henkel. Trace-driven system-level power evaluation of system-on-a-chip peripheral cores. In *Asia South-Pacific Design Automation Conference (ASP-DAC)*, Jan. 2001.

[10] V. Pareto. *Cours D'Economie Politique*, volume I–II. Lausanne, 1896.

[11] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In L. Erlbaum, editor, *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, 1985.

[12] F. Vahid and T. Givargis. The case for a configure-and-execute paradigm. In *International Workshop on Hardware/Software Codesign (CODES)*, pages 59–63, May 1999.

[13] D. A. V. Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.

[14] M. Wall. *GAlib: A C++ Library of Genetic Algorithm Components*. Mechanical Engineering Department, Massachusetts Institute of Technology, Aug. 1996.

[15] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[16] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 4(3):257–271, Nov. 1999.

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 **IEEE**