A Novel Approach to Design Space Exploration of Parameterized SOCs

Giuseppe Ascia Vincenzo Catania
Maurizio Palesi
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Università di Catania, Italy
{gascia,vcatania,mpalesi}@diit.unict.it

Abstract

The paper proposes a methodology based on evolutionary techniques for exploration of the range of possible configurations of a parameterized system [8]. A highly parametric system-on-a-chip for digital camera applications will be taken as a case study and a multi-objective genetic algorithm will be used to search for the power-performance trade-off surface. The methodology proposed will be compared with that implemented in Platune [1] in terms of both accuracy and efficiency in relation to the number of simulations performed.

1. Introduction

A recent reduction in the time to market has led to the development of a new approach to 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* [13] 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 [2] 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 [9] 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 [10] the same authors describe a method 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-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 stated in formal terms and an overview of multi-objective optimisation techniques based on genetic algorithms will be given. Section 3. will present the methodology used to explore the

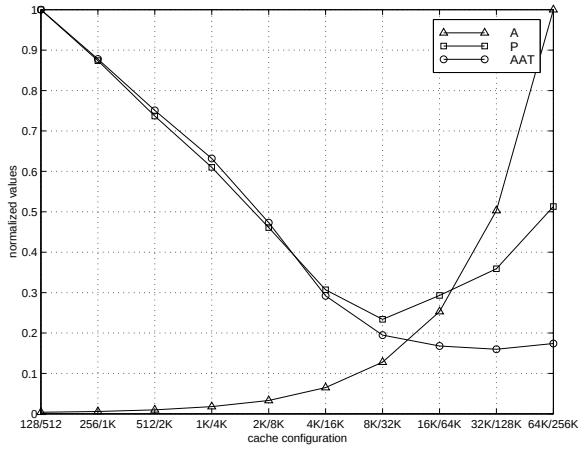

Figure 1: Normalised trends for area, total switching capacitance and average access time for various cache canfigurations.

range of configurations and apply it to a case study. Section 4. will present 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. For example, Figure 1 shows the normalised trends for area (A), total switching capacitance (P) and average access time (AAT) for a three-level memory hierarchy with various cache sizes. In each configuration the first-level caches are of the same size while the second-level cache is 4 times the size of the first-level ones. It is clearly impossible to find a configuration that optimises all the objectives at the same time.

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 [3]: *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 $\overline{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which will satisfy the m inequality constraints:

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

the p equality constraints

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

and optimizes the vector function

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

where $\overline{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

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

Multiobjective optimisation dates back to Vilfredo Pareto's treatise on political economy in [11]. We say that a point $\overline{x}^* \in \mathcal{F}$ is Pareto optimal if for every $\overline{x} \in \mathcal{F}$ either,

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

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

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

This definition states that $\overline{x}^*$ is Pareto optimal if there exist no vectors $\overline{x} \in \mathcal{F}$ 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. Methodology Proposed

An exhaustive search for the optimal configurations of a parameterized system is computationally unfeasible. Evaluation of a configuration requires simulation of the system once configured. Even if a high-level model of the system is available, thus allowing for rapid simulation of a configuration and estimation with a reasonable degree of accuracy of the variables to be optimised, it would be unthinkable to evaluate the whole range of configurations in order to find the Pareto optimal-set.

One possible approach to exploring the range of configurations uses heuristic techniques to limit the range [6, 14]. The main disadvantage of this approach

is that it requires accurate analysis of the architecture to identify and discard any Pareto-dominated configurations and thus avoid the need to simulate them. This can be solved by using evolutionary techniques and thus treating the exploration in terms of a problem of global optimisation [2].

Application of evolutionary algorithms (EAs) in multiobjective optimisation has attracted the attention of researchers from different backgrounds [5]. GA-based approaches to multiobjective optimisation are divided into two classes: those not based on the notion of Pareto optimum and Pareto based ones. The first class includes approaches that use aggregating functions to transform the problem of multiobjective optimisation into one of scalar optimisation [15, 12]. This approach was used in [2] to search for the configurations that minimised a cost function defined as the aggregate of the area, power and average access time parameters in a memory hierarchy. The main disadvantage of approaches based on 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. This can be solved by using Pareto-based approaches in which the idea is to find the individuals that are Pareto non-dominated by the rest of the population. These individuals are then assigned the highest rank and eliminated from further contention. Another set of Pareto non-dominated individuals are determined from the remaining population and are assigned the next highest rank. The procedure is repeated until the whole population is suitably ranked.

In this paper we have considered multiobjective optimisation techniques that use Pareto-based GAs. More specifically, we chose the *Strength Pareto EA* (SPEA) [18, 17] 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 2. 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.

The algorithm was implemented using GAlib [16] (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 [4] as rejection of unfeasible individuals.

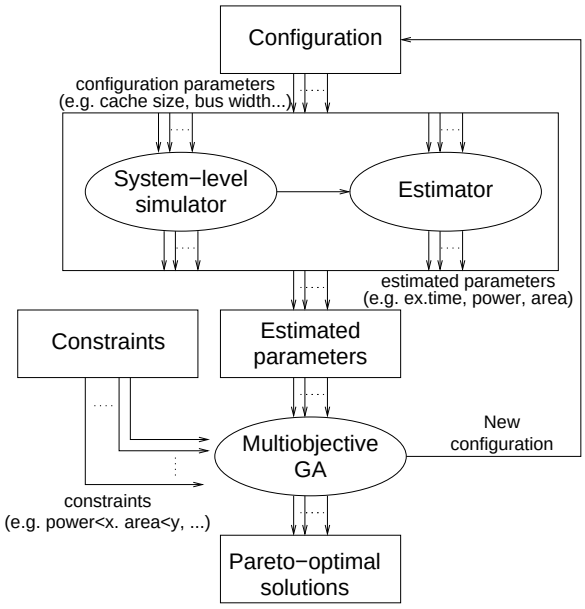

Figure 2: Design flow.

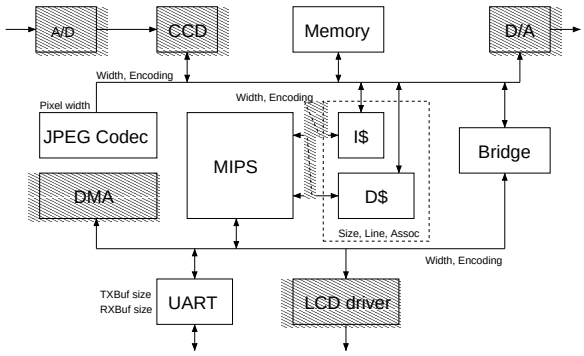

Figure 3: Reference architecture.

The methodology was applied to the architecture shown in Figure 3. It is a highly parametric SOC for digital camera applications developed under 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.

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/abus width	$4, 8, \ldots, 32$	$4 \times 2 \times 2$
	dbus/abus encoding	$bin, gray, inv$	$3 \times 2 \times 2$
\$ $\leftrightarrow$ MEM	dbus/abus width	$4, 8, \ldots, 32$	4×2
	dbus/abus encoding	$bin, gray, inv$	3×2
Peripheral bus	dbus/abus width	$4, 8, 16, 32$	4×2
	dbus/abus encoding	$bin, gray, inv$	3×2
UART	TX/RX buf size	$1, 2, 4, 8, 16$	5×2
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.

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 [9].

4. Results

A good multiobjective optimisation algorithm has to be able to find a trade-off surface as close as possible to the Pareto-optimal front, evaluating as few configurations as possible. As mentioned previously, evaluation of a configuration means simulating the system to estimate the variables to be optimised. Simulation of a complex system like a SOC is an extremely expensive operation in terms of CPU time, so reducing the number of simulations is a key issue. For this reason the methodology proposed was validated in terms of both accuracy in the search for the Pareto-optimal set and efficiency as regards CPU time.

The approach based on evolutionary techniques was compared with the configuration space exploration algorithm implemented in Platune [1], in which analysis of the parameters and dependencies showed that a specific configuration order cuts out several of the non-Pareto optimal points and thus generates all the Pareto-optimal points in a reasonable amount of time [14]. The graph in Figure 4 shows execution time versus power absorbed during execution of a program that shifts a bitmap from one memory region to another. The points shown refer to the Pareto-set obtained with Platune and our approach for varying numbers of generations. Of course, the larger the number of generations through which the GA is made to evolve, the closer the Pareto-set will be to the Pareto-optimal set. On the other hand, a larger number of generations means more evaluations, so a trade-off has to be reached between accuracy and efficiency. Figure 5 compares the two approaches in terms of efficiency, giving the total number of evaluations made and the percent saving obtained by the genetic approach as compared with Platune for varying numbers of generations. As can be seen, the results obtained are very close to those provided by Platune but with about 60% fewer evaluations.

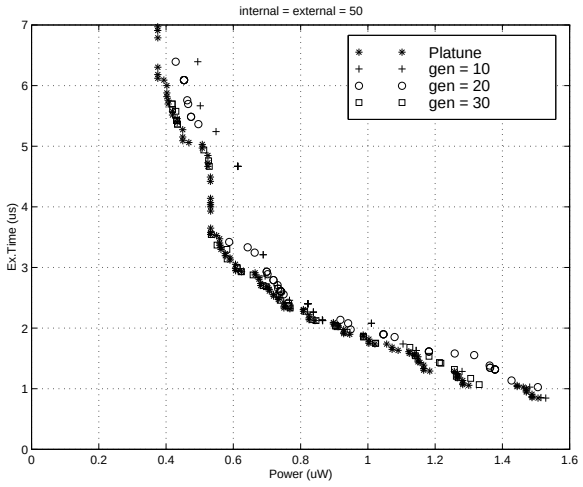

Figure 4: Trade-off surface for various generations (internal population size = external population size = 50).

If the size of the population is increased, even better results are obtained but the number of evaluations increases as well. Figure 6 gives the results for an external and internal population of 100 individuals and varying numbers of generations. As can be seen, the number of points found is higher than in the previous case, although the number of evaluations required is only slightly higher and at any rate about 50% of the number performed by Platune (see Figure 7).

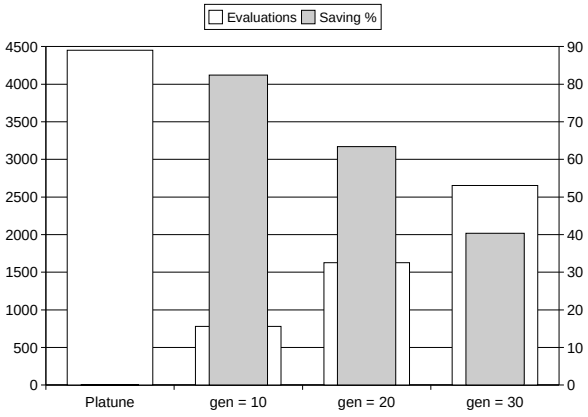

Figure 5: Total number of evaluations made and percent saving obtained by the genetic approach as compared with Platune for varying numbers of generations (internal population size = external population size = 50).

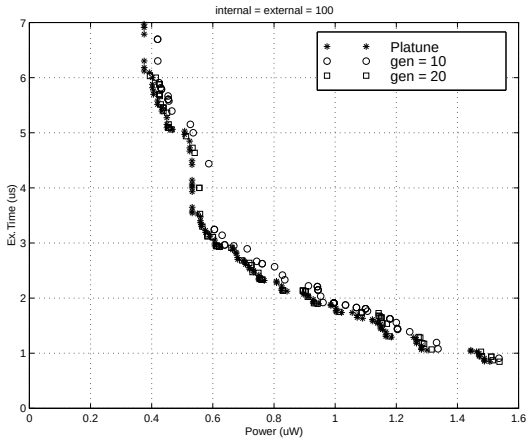

Figure 6: Trade-off surface for various generations (internal population size = external population size = 100).

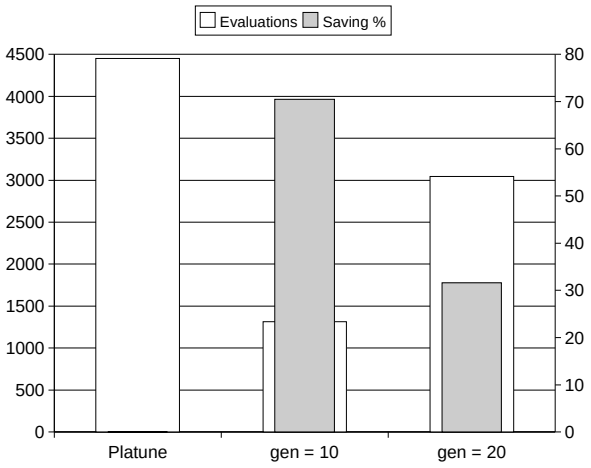

Figure 7: Total number of evaluations made and percent saving obtained by the genetic approach as compared with Platune for varying numbers of generations (internal population size = external population size = 100).

5. Conclusions

In this paper we have proposed a GA-based methodology for multiobjective exploration of the range of configurations for a parameterized system. The methodology was applied to a highly parametric SOC for digital camera applications. The approach was evaluated in terms of both accuracy and the CPU time required to explore the range of configurations and compared with the approach used in Platune [1]. The results obtained show that unlike others this approach solves the problem by successive refinement: the more the algorithm is made to evolve, the closer the Pareto-set found is to the Pareto-optimal set. This would appear to be a very useful feature, as the user can choose the accuracy/CPU time trade-off. In terms of CPU time to search for the trade-off front, the approach requires on average 60% fewer simulations than the Platune approach.

Future developments will address definition of a mixed genetic/heuristic approach using evolutionary techniques to achieve global optimisation together with efficient local optimisation techniques.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[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 and F. Vahid. Parametrized system design. In *8th International Workshop on Hardware/Software Codesign*, 2000.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] F. Vahid and T. Givargis. Platform tuning for embedded systems design. *IEEE Computer*, 34(3):112–114, Mar. 2001.

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

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

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

[18] 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.