DESIGN SPACE EXPLORATION METHODOLOGIES FOR IP-BASED SYSTEM-ON-A-CHIP

Giuseppe Ascia, Vincenzo Catania, Maurizio Palesi

Università di Catania
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
V.le Andrea Doria, 6 — 95125 Catania, Italy

ABSTRACT

In this paper we present two new solutions for design space exploration of parameterized systems. The approaches use multi-objective optimisation techniques based on the concept of Pareto-optimality to determine the power/performance trade-off front for a highly parameterized system-on-a-chip for digital camera applications. The approaches used are purely heuristic and a combination of heuristic approach with genetic algorithm-based approach. The results obtained demonstrate the effectiveness of the approaches in terms of both validity and efficiency, measured as the number of simulations run.

1. INTRODUCTION

The increasing technological gap between the number of transistors that could be used and the number actually used [1] has partially been solved by the IP-based design approach that uses pre-designed and pre-verified cores as building blocks (as gates were previously used). In this field a new approach, called *configure-and-execute* was proposed in [2] and is based on the presence of a highly parametric pre-designed system-on-a-chip or (SOC platform) which is configured according to the application or set of applications it will have to execute. The parameter configuration optimises an objective function that almost always depends on three variables—area, power and performance.

Various approaches to explore the range of configurations have been proposed. In [3] 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. In [4] sensitivity analysis was used to search for the configuration that minimises the power-delay product for a cache memory. In [5] multi-objective genetic algorithms (GAs) were used to search for optimal configurations in terms of area, power and average access time for a system-on-chip.

This paper presents two new approaches to design space exploration (DSE) for parameterized systems. The approaches use multi-objective optimisation techniques based on the concept of Pareto optimality [6]. The first approach is an extension of sensitivity analysis [4] to multi-objective optimisation, the second uses a combination of the previus one with multi-objective evolutionary programming techniques. The results obtained in a case study (a highly parameterized SOC for digital camera applications) show the effectiveness of the techniques proposed for DSE in terms of both accuracy and efficiency, measured as the number of simulations run.

The paper is structured as follows. In Section 2 two new approaches to design space exploration are presented. In Section 3 some experimental results are described. Finally, Section 4 provides our conclusions.

2. MULTI-OBJECTIVE METHODOLOGIES FOR DESIGN SPACE EXPLORATION (DSE)

One of most important problem to solve in system level design of a IP-based parameterized system is the definition of a methodology to generate the Pareto-optimal set of configurations that optimise towards several objectives. Evaluation of a generic configuration requires a 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 for complex systems 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 reduce the space of configurations that have to be analysed by identifying and discarding any Pareto-dominated [4] configurations. Another approach presented in [7] 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 [8].

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.

0-7803-7448-7/02/$17.00 ©2002 IEEE

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.

2.1. Pareto-Based Sensitivity Analysis

The sensitivity analysis (SA) methodology presented in [4] divides the space exploration of possible configurations in two phases. The first phase identifies 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 propose an extension of the sensitivity analysis methodology to perform multi-objective optimisation based on the notion of Pareto optimum (PBSA).

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

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

where dist is the Euclidean distance and $\bar{o}_j^{(i)}$ 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, whose components represent the objective values, the non-dominated configurations are extracted and accumulated in a set $\mathcal{P}$. At the second iteration for each configuration in the set $\mathcal{P}$, $p_{S_2} \in \mathcal{V}^{(S_2)}$ is made to vary. From the $N_{S_2} \times |\mathcal{P}|$ obtained, the non-dominated configurations are extracted and accumulated in the set $\mathcal{P}$. The procedure is repeated for all the parameters S_i whose sensitivity s_{S_i} exceeds a certain threshold whose value is fixed by the designer. As the other parameters have a restricted influence on the value of the objective functions, they are set to a reference value (e.g. the mean value of their variation set). At the end of the algorithm the configurations in $\mathcal{P}$ 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
  $\mathcal{ND} = \{\bar{p}^*\}$ // initialize non-dominated set
  $i = 1$ // high sensitive parameter index
  repeat
    $\mathcal{C} = \{\}$
    for all $\bar{c} \in \mathcal{ND}$ **do**
      for all $v \in \mathcal{V}^{(S_i)}$ **do**
        $\bar{c}[S_i] = v$
        $\mathcal{C} = \mathcal{C} \bigcup \{\bar{c}\}$
      end for
    end for
    $\mathcal{ND} = \mathcal{ND} \bigcup \mathcal{C}$
    $\mathcal{ND} = \mathcal{ND} \setminus \text{Dominated}(\mathcal{ND})$ // remove dominated solutions
    $i = i + 1$ // next high sensitive parameter
  until $i > P$ OR Sensitivity$(S_i) <$ MINSENS

2.2. Sensitivity Analysis Genetic Algorithm

In [7] a multi-objective methodology based on Pareto-based Genetic Algorithm was proposed. This methodology allows a rapid exploration of the space of configurations and

it is very effective in sampling from along the entire Pareto-optimal front and distributing the solutions generated over the trade-off surface. As any hypothesis on the system is required, this methodology uses all the parameters of the system to define the chromosome of the genetic algorithm. The sensitivity analysis approach, on the other hand, allows reduction of the configuration space to be evaluated by neglecting parameters that have less effect on the objective functions. A mixed approach we propose to exploit the potential of the previous two approaches. It is based on the multiobjective genetic approach using only the most sensitive parameters determined by the sensitivity analysis. We will call this new mixed approach Sensitivity Analysis Genetic Algorithm (SAGA).

Obviously the trade-off front obtained by SAGA will not be better than that obtained by the pure GA approach, given that the space of configurations on which SAGA operates is a subspace of the space on which the GA approach operates. As compared with PBSA, on the other hand, the solutions found will be better, as the parameter tuning is not constrained: any combination of parameter values is admissible.

The flow of operations performed by SAGA, is divided into two stages. In the first, the sensitivity analysis is performed to determine the sensitivity of each parameter. The parameters S_i whose sensitivity s_{S_i} are greater than a certain sensitivity threshold are used to define the chromosome of the genetic algorithm that will be applied in the second stage. The remaining parameters that remain below the threshold are set to a reference value (e.g. the mean value of their range of variation).

3. APPLICATION OF THE PROPOSED METHODOLOGIES TO A CASE STUDY

The reference architecture we used to test the methodologies proposed in Section 2 is Figure 1. It is a highly parameterized SOC for digital camera applications [9]. The system is 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. 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

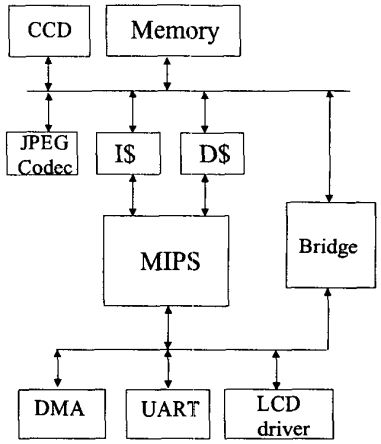

Fig. 1. Reference architecture

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 [10].

The two proposed approaches were compared with pure genetic algorithm approach 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 2 gives the power/execution-time trade-off for the *image* application, obtained using the three approaches. PBSA was executed with different threshold values. As was to be expected, when the threshold decreases more parameters are taken into consideration, thus leading to an improvement in the solutions obtained but an increase in the number of simulations required (respectively 1780, 4958 and 8633 for thresholds of 10%, 5% and 1%). SAGA gives the same results as PBSA with a 1% threshold but after only 30 generations with internal and external populations of 50 individuals, and a total of 2238 simulations. With GA, after 50 generations with internal and external populations of 50 individuals, a total of 4581 simulations gives dominant solutions as compared to those obtained with PBSA and SAGA.

The same qualitative results are obtained in the other two applications – *key* and *matrix* and Table 1 gives the number of simulations run by each approach and the percent gain over PBSA. The results were obtained using a 1% threshold for both PBSA and SAGA. For SAGA and GA we used an internal and external population of 50, a crossover probability of 0.9 and a mutation probability of 0.01. GA and SAGA respectively converged after 50 and 30 generations for *Image* and *Key* and after 40 and 20 generations for *Matrix*.

In short, the solutions obtained with GA dominate those obtained with PBSA and are achieved with on average 46% fewer simulations. If we are willing to sacrifice the quality

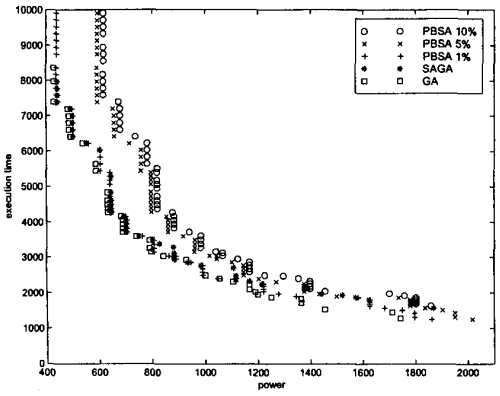

Fig. 2. Power/execution-time trade-off for the *image* application

Application	PBSA	GA		SAGA	
		sims	saving %	sims	saving %
Image	8633	4581	47.5	2238	74.1
Key	11019	4587	58.4	2642	76
Matrix	4372	3751	14.2	1714	60.8
Total	24024	12919	46.2	6594	72.6

Table 1. Number of simulations run by each approach and the percent gain over PBSA

of the solutions to obtain an increase in efficiency, we can use the mixed SAGA approach, which gives solutions close to those obtained with GA but with 72% fewer simulations than PBSA.

4. CONCLUSIONS

In this paper we have presented two new approaches to design space exploration for parameterized systems: the first approach is heuristic and is based on sensitivity analysis, the second is a mixture of the sensitivity analysis with multi-objective genetic algorithms. All these approaches were applied to determine the power/ performance trade-off of a highly parameterized architecture implementing a SOC for digital camera applications during the execution of different applications. The approaches were evaluated in terms of both the quality of the solutions obtained (using the concept of Pareto dominance) and efficiency, measured as the number of simulations required to determine the trade-off front.

The results obtained show the effectiveness of the pure genetic approach as regards the quality of the solutions obtained, and the mixed approach in terms of efficiency.

5. REFERENCES

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

[2] Frank Vahid and Tony Givargis, "The case for a configure-and-execute paradigm," in *International Workshop on Hardware/Software Codesign (CODES)*, May 1999, pp. 59–63.

[3] Tony Givargis, Jörg Henkel, and Frank Vahid, "Interface and cache power exploration for core-based embedded system design," in *International Conference on Computer-Aided Design (ICCAD)*, Nov. 1999, pp. 270–273.

[4] William Fornaciari, Donatella Sciuto, Cristina Silvano, and Vittorio Zaccaria, "A design framework to efficiently explore energy-delay tradeoffs," in *9th. International Symposium on Hardware/Software Co-Design*, Copenhagen, Denmark, Apr. 25–27 2001, pp. 260–265.

[5] Giuseppe Ascia, Vincenzo Catania, and Maurizio Palesi, "Parameterized system design based on genetic algorithms," in *9th. International Symposium on Hardware/Software Co-Design*, Copenhagen, Denmark, Apr. 25–27 2001, pp. 177–182.

[6] Vilfredo Pareto, *Cours D'Economie Politique*, vol. I–II, Lausanne, 1896.

[7] Giuseppe Ascia, Vincenzo Catania, and Maurizio Palesi, "A novel approach to design space exploration of parameterized socs," in *11th. IFIP International Conference on Very Large Scale Integration*, Montpellier, France, Dec. 3-5 2001, pp. 449–456.

[8] David A. Van Veldhuizen and Gary B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.

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

[10] Tony Givargis, Frank Vahid, and Jörg Henkel, "A hybrid approach for core-based system-level power modeling," in *Asia and South Pacific Design Automation Conference*, 2000.