

A Flexible Framework for Fast Multi-Objective Design Space Exploration of Embedded Systems

Gianluca Palermo¹ Cristina Silvano¹ Vittorio Zaccaria²

¹ DEI, Politecnico di Milano, Milano Italy
{gpalermo, silvano}@elet.polimi.it

² STMicroelectronics, Agrate Brianza, Italy
{vittorio.zaccaria}@st.com

Abstract. The evaluation of the best system-level architecture in terms of energy and performance is of mainly importance for a broad range of embedded SOC platforms. In this paper, we address the problem of the efficient exploration of the architectural design space for parameterized microprocessor-based systems. The architectural design space is multi-objective, so our aim is to find all the Pareto-optimal configurations representing the best power-performance design trade-offs by varying the architectural parameters of the target system. In particular, the paper presents a Design Space Exploration (DSE) framework tuned to efficiently derive Pareto-optimal curves. The main characteristics of the proposed framework consist of its flexibility and modularity, mainly in terms of target architecture, related system-level executable models, exploration algorithms and system-level metrics. The analysis of the proposed framework has been carried out for a parameterized superscalar architecture executing a selected set of benchmarks. The reported results have shown a reduction of the simulation time of up to three orders of magnitude with respect to the full search strategy, while maintaining a good level of accuracy (under 4% on average).

1 Introduction

Decreasing energy consumption without a relevant impact on performance is a 'must' during the design of a broad range of embedded applications. Evaluation of energy-delay metrics at the system-level is of fundamental importance for embedded applications characterized by low-power and high-performance requirements. The growing diffusion of SOC embedded applications based on the platform-based design approach requires a flexible tuning framework to assist the

phase of *Design Space Exploration (DSE)*. In general, different applications could impose different energy and performance requirements. The overall goal of the DSE phase is to optimally configure the parameterized SOC platform in terms of both energy and performance requirements depending on the given application. In general, parameterized embedded System-On-Chip architectures must be optimally tuned to find the best energy-delay trade-offs for the given classes of applications. The value assignment to each one of the system-level parameters can significantly impact the overall performance and power consumption of the given embedded architecture. To explore the large design space for the target architecture, an approach based on the full search of the optimal architectural parameters at the system-level with respect to the energy-delay cost function can be computationally very costly due to the long simulation time required to explore the wide space of parameters.

The problem addressed in this paper consists of defining a flexible Design Space Exploration (DSE) framework to efficiently explore the multi-objective design space in order to find a good approximation of Pareto-optimal curves representing the best compromise between the interesting design objectives, mainly energy and delay. The proposed DSE framework is flexible and modular in terms of: target architecture and related system-level executable models, exploration algorithms, and system-level metrics. The target SOC platform consists of a parameterized superscalar architecture. The set of tunable parameters is mainly related to the target microprocessor, the memory hierarchy and the system-level interconnection buses. Each parameterized component of the target architecture is provided with the corresponding system-level executable model to dynamically profile the given application to derive the information related to power and performance metrics.

The main goal of the selected set of exploration algorithms is its efficiency with respect to the full search exploration algorithm, while preserving a good level of accuracy. Up to now, the exploration algorithms plugged in the DSE framework are the Random Search Pareto (RSP) technique, the Pareto Simulated Annealing (PSA), and the Pareto Reactive Tabu Search (PRTS). The algorithms have been tuned to efficiently derive a good approximation of Pareto-optimal curves. The analysis of the proposed framework has been carried out for a parameterized superscalar architecture executing a set of multimedia applications. The reported results have shown a reduction of the simulation time of up to three orders of magnitude with respect to full search strategy, while maintaining a good level of accuracy (under 4% on average).

The rest of the paper is organized as follows. A review of the most significant works appeared in literature concerning the DSE problem is reported in Section 2. The DSE problem is stated in Section 3 along with the problem of the approximation of Pareto curves. The proposed DSE framework is described in Section 4, while Section 5 discusses the experimental results carried out to evaluate the efficiency of the proposed framework for a superscalar configurable target architecture. Finally some concluding remarks have been reported in Section 6.

2 Background

Several system-level estimation and exploration methods have been recently proposed in literature targeting power-performance tradeoffs from the system-level standpoint [1], [2], [3], [4], [5], [6], [7].

The SimpleScalar toolset [5] is based on a set of MIPS-based architectural simulators focusing on different abstraction levels to evaluate the effects of some high-level algorithmic, architectural and compilation trade-offs. The SimpleScalar framework provides the basic simulation-based infrastructure to explore both processor architectures and memory subsystems. However, SimpleScalar does not support power analysis. Based on the SimpleScalar simulators, SimplePower [8] can be considered one of the first efforts to evaluate the different contributions to the energy budget at the system-level. The SimplePower energy estimation environment consists of a compilation framework and an energy simulator that captures the cycle-accurated energy consumed by the SimpleScalar architecture, the memory system and the buses. More recently, the Wattch architectural-level framework has been proposed in [6] to analyze power with respect to performance tradeoffs with a good level of accuracy with respect to lower-level estimation approaches. Wattch represents an extension of the SimpleScalar simulators to support power analysis at the architectural level. Wattch provides a framework to explore different system configurations and optimization strategies to save power, in particular focusing on processor and memory subsystems.

The *Avalanche* framework [2] evaluates simultaneously the energy-performance tradeoffs for software, memory and hardware for embedded systems. The *Avalanche* framework mainly focuses on the processor and memory subsystems. The work in [4] proposes a system-level technique to find low-power high-performance superscalar processors tailored to specific user applications. Low-power design optimization techniques for high-performance processors have been investigated in [7] from the architectural and compiler standpoints. A trade-off analysis of power performance effects of SOC (System-On-Chip) architectures has been recently presented in [9], where the authors propose a simulation-based approach to configure the parameters related to the caches and the buses.

Among DSE framework appeared in literature so far, very few approaches have been introduced recently to approximate Pareto-curve construction for computer architecture design [9] [10].

In general, the most trivial approach to determine the Pareto-optimal configurations into a large design space with respect to a multi-objective design optimization criteria consists of the comprehensive exploration of the configuration space. This *brute force* approach can be feasible only if the number of parameters in the configuration space is very limited. On the contrary, it is quite common to find a design space composed of tens of parameters, leading to an exponential analysis time. Thus, traditional heuristics must be used. When the design space is too large to be exhaustively explored, heuristic methods must be adopted to find acceptable near-optimal solutions. The problem of the efficient construction of Pareto curves has been often addressed by using domain-specific algorithms.

For example, the high-level synthesis scheduling problem (minimization of latency with area constraints) implicitly needs to consider an approximation of the Pareto curve of the area/latency design evaluation space. For this problem, specific approaches have been proposed in the past [11]. However, due to the high generality of the design space of a platform design, domain-specific algorithms are very difficult to find, thus one should resort to traditional heuristics.

Platune [9] is an optimization framework that exploits the concept of parameter independence to individuate approximate Pareto curves without performing the exhaustive search over the whole design space. The authors define two parameters as interdependent if changing the value of one of them impacts the optimal parameter value of the other. In this case, all the combinations of these two parameters must be analyzed to find an optimal configuration. However, if the parameters are independent, the two subspaces can be analyzed separately, leading to a reduced simulation time. The main drawback of this approach is that parameter independence must be specified by the user by means of a dependency graph since no automatic methods are proposed for such task. Platune is not a modular framework since it allows only the exploration of a MIPS based system and its goodness has not been compared with simpler approaches such as random search or full parameter space exploration.

More recently, Palesi et al. [10] extended Platune by applying genetic algorithms to optimize dependent parameters, resorting to the default Platune policy when independent parameters are specified by the user. However, their approach is always based on an a-priori parameter dependency graph to be given by the user and it is compared only with the default Platune policy.

3 Design Space Exploration

To meet the time-to-market constraints, modern design techniques oriented to the reuse of intellectual properties are increasing their importance. For example, the use of a customizable System-On-Chip (SOC) platforms [12] where a stable microprocessor-based architecture can be easily extended and customized for a range of applications, enable a quick deployment and low cost high level design flow. For example, even considering a simple embedded microprocessor-based architecture composed of the CPU and the memory hierarchy, the identification of the optimal system configuration by trading off power and performance still leads to the analysis of too many alternatives. The overall goal of this work aims at overcoming such problems by providing a methodology and a design framework to drive the designer towards optimal solutions in a cost-effective manner.

3.1 Parameterized Design Space Definition

We define the *design space* as the set of all the feasible architectural implementations of a platform. A possible configuration of the target architecture is

mapped to a generic point in the design space as the vector $a \in \mathcal{A}$ where \mathcal{A} is the architectural space defined as:

$$\mathcal{A} = S_{p_1} \times \dots \times S_{p_i} \dots \times S_{p_n}$$

where S_{p_i} is the ordered set of possible configurations for parameter p_i and "×" is the cartesian product. Associated with each point a of the design space, there are a set of evaluation functions (or *metrics*). The *design evaluation space* is the multi-dimensional space spanned by these evaluation functions.

The problem afforded in this paper consists of platform optimization, that is searching for the best design, i.e., an implementation that optimizes all the objectives within the design evaluation space. However, the optimization problem involves the minimization (maximization) of *multiple objectives* making the definition of optimality not unique. To address this problem, let us introduce the definition of *Pareto point* for a minimization problem [13]. A Pareto point is a point of the design space, for which there is no other point with at least an inferior objective, all others being inferior or equal. Obviously, a Pareto point is a global optimum in the case of a monodimensional design space, while in the case of multi-dimensional design evaluation space, the Pareto points form a trade-off curve or surface called Pareto curve.

In general, Pareto points are solutions to *multi-objective* or *constrained* optimization problems. For example, we can be interested in minimizing the power consumption under a delay constraint or viceversa. The solution of this problem is straightforward if the Pareto curve is available. However, a Pareto curve for a specific platform is available only when all the points in the design space have been characterized in terms of objective functions. This is often unfeasible due to the cardinality of the design space and to the long simulation time needed for computing the evaluation functions.

Our target problem consists of finding a good approximation of the Pareto curves by trading off the accuracy of the approximations and the time needed for their construction. In a system-level description, the designer is required to specify the boundaries of the design space for each parameter (for example, the maximum size to be considered for the cache size) and other possible freezing constraints on some parameters.

4 Proposed Design Space Exploration Framework

The overall goal of this work aims at providing a methodology and a retargetable tool to drive the designer towards near-optimal solutions, with the given multiple constraints, in a cost-effective fashion. The final product of the framework is a Pareto curve of configurations within the design evaluation space of the given application. To meet our goal we implemented a skeleton for an extendible and easy to use framework for multi-objective exploration. The proposed DSE framework is flexible and modular in terms of:

- target architecture and related system-level executable models;

- exploration algorithms;
- system-level metrics.

Given a new target architecture, the user can plug in the new architecture module in the framework, while the other modules can be considered as ready-to-use black boxes. Similarly, given a new exploration algorithm or a new executable module, the user can plug-in the new module in the framework.

The proposed framework is shown in Figure 1. It is mainly composed in two modules: System Description Module (SDM) and Design Space Exploration Module (DSEM). The DSEM receives as input the description of the possible design configurations (i.e. the target *design space*) and the application for which the system has been designed and for which the optimal configuration must be found (SDM). The framework explores the design space by using an iterative optimization technique with respect to different metrics.

The *Optimizer* module is responsible for choosing, from the design space, a set of candidate optimal points to be evaluated. Once selected, each point is mapped into a specific instance of the target architecture by the "*Architecture Mapping*" module. Providing a 'mapping' between the high-level description interface used by the Optimizer and the actual specification of the target architecture, this module enables the evaluation of each point by means of simulation based on the executable model of the target system. In the presented framework, the model of the target architecture can be described at different abstraction levels, being available the corresponding simulator.

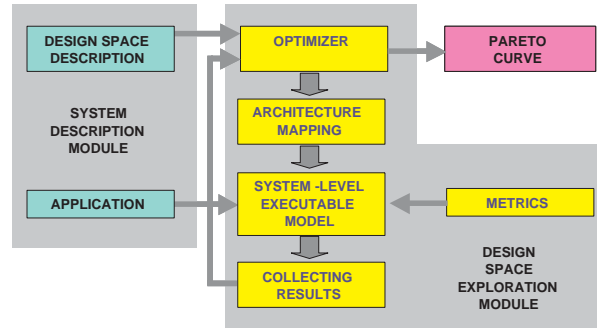


Fig. 1. Proposed design space exploration framework

Up to now, the approximation of the Pareto curve could be performed by a set of already plugged in exploration algorithms:

- Random Search Pareto(RSP) [14]. RSP algorithm is derived from *Monte Carlo* methods. In general, the main characteristic of *Monte Carlo* methods is the use of random sampling techniques to come up with a solution of the target problem. The random sampling technique has been proved to be one of the best techniques to avoid falling into local minima.

- Pareto Simulated Annealing (PSA) [15]. Simulated annealing is a Monte Carlo approach for minimizing such multivariate functions. The term simulated annealing derives from the analogy with the physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure. In the Simulated Annealing algorithm a new configuration is constructed by imposing a random displacement. If the cost function of this new state is less than the previous one, the change is accepted unconditionally and the system is updated. If the cost function is greater, the new configuration is accepted probabilistically; the acceptance possibility decreases with the temperature. This procedure allows the system to move consistently towards lower cost function states, thus 'jumping' out of local minima due to the probabilistic acceptance of some upward moves. The PSA is an evolution of SA for multi-objective optimization. At each step of PSA, the starting point is not a single configuration but a set of configurations (*Partial Pareto Set*).
- Pareto Reactive Tabu Search (PRTS) [16]. The Tabu Search (TS) is an iterative algorithm that explores the design space by means of 'moves'. The key concept behind the algorithm is the tabu list, i.e., a list containing prohibited moves that, usually, consist of the most recently visited points. The reason of the tabu list is to avoid to stuck into local minima. Recent studies [16] have demonstrated that the length of this list is a determining factor to reduce the possibility to stuck into local minima and a careful tuning of the list length is of fundamental importance for the success of the algorithm. The Reactive Tabu Search is an evolution of the Tabu Search algorithm that exploits an *adaptive prohibition period*, paired with an *escape mechanism*, to afford the tuning problem. In RTS, the prohibition period of a specific solution increases with the frequency of the visits to that solution. Moreover, to avoid the possibility of a cyclic exploration, an escape mechanism is used in order to escape from local minimum. The escape is usually implemented by generating a random walk.

Our effort has been devoted to the tuning phase of the parameters requested in the described set of algorithms.

5 Experimental Results

In this section, we present the experimental results obtained by applying the proposed DSE framework to optimize a superscalar microprocessor-based system. The first subsection of this paragraph describes the target architecture and the related design evaluation space; the second subsection discusses the application of the framework to two case studies.

5.1 Target System Architecture

In general, a superscalar architecture is composed of many parameters, so that the design space to explore is quite large. Our analysis has been focused on

those design parameters significantly impacting the performance and the energy consumption. Each instance of the virtual architecture has been described in terms of the following parameters:

- $\mathbf{S}_{s_i}, \mathbf{S}_{s_d}, \mathbf{S}_{s_{u2}}$ are the ordered sets of the possible sizes of the I/D L1 caches (from 2 KByte to 16 KByte) and unified L2 cache (from 16 KByte to 128 KByte).
- $\mathbf{S}_{b_i}, \mathbf{S}_{b_d}, \mathbf{S}_{b_{u2}}$ are the ordered sets of the possible block sizes of the I/D L1 caches (from 16 Byte to 32 Byte) and unified L2 cache (from 32 Byte to 64 Byte).
- $\mathbf{S}_{a_i}, \mathbf{S}_{a_d}, \mathbf{S}_{a_{u2}}$ are the ordered sets of the possible associativity values of the I/D L1 caches (from 1 way to 2 ways for the I-cache and from 2 ways to 4 ways for the D-cache) and unified L2 cache (from 4 ways to 8 ways).
- $\mathbf{S}_{ia}, \mathbf{S}_{im}$ are the ordered sets of the possible number of integer ALUs and multipliers (from 1 to 2).
- $\mathbf{S}_{fpa}, \mathbf{S}_{fpm}$ are the ordered sets of the possible number of floating point ALUs and multipliers (from 1 to 2).
- \mathbf{S}_{iw} are the ordered sets of the possible issue width sizes (from 2 to 8).

Wattch simulator [6] has been used as our target architectural simulator providing a dynamic profiling of energy and delay.

5.2 Application of the Methodology

In this subsection, we report the results in terms of the efficiency and accuracy of the application of our DSE methodology to a set of benchmarks composed of a set of DCT transforms and FIR filters as well as other numerical algorithms written in C language.

To validate our exploration methodology we carried out two parallel exploration flows: the first is based on the exhaustive search, while the second is based on the given algorithm. Each benchmark has been optimized independently with the two flows and the resulting Pareto curves have been compared. In the full search case, the optimizer analyzes the global design space, so the number of simulations to be executed is 196608. This corresponds to approximately 370 hours of simulations for the selected set of benchmarks. Table 1 shows, in the first row, the number of visited configurations ($0/00$) with respect to the total number of configurations. In the second row, Table 1 shows the average percentage error of the approximate Pareto curve obtained with the PSA algorithm with respect to the exhaustive search. To evaluate the accuracy of the approximate curve we used the error defined in [omitted for blind review] Table 1 shows a fast convergence for the PSA algorithm due to an accurate tuning of its parameters: we found an average error under the 4% reducing up to three orders of magnitude the simulation time.

Figure 2 presents the results obtained by applying the proposed framework to two benchmarks: the FDCT and FIR1. In this case, the optimization module used to find the Pareto curves is the PSA algorithm. Figure 2 shows the scatter

plot of all the configurations generated in the energy-delay space by the PSA algorithm (light gray) and the corresponding Pareto configurations (dark gray). Once the Pareto curve has been found, a constrained design space exploration problem could select its solution among the Pareto points. For our two cases studies based on the PSA algorithm, we found the approximated Pareto curve in approximately one hour of simulation with respect to a week requested by the exhaustive search.

Table 1. Efficiency and accuracy of the PSA-based exploration vs. full search

Visited Configuration [‰]	0.5	1	5	10	50
Average Error [%]	6	4	2	1	0.5

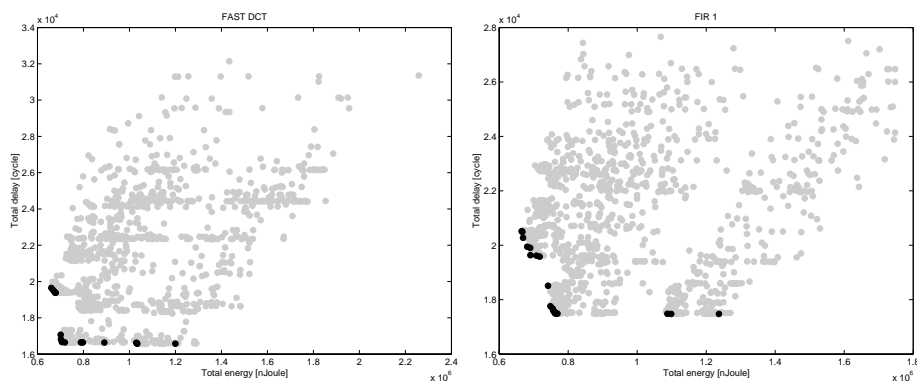


Fig. 2. Energy-delay PSA-based exploration results for FAST DCT and FIR1

6 Conclusions

In this paper, a flexible design space exploration framework has been proposed to efficiently derive Pareto-optimal curves. The framework is completely configurable in terms of target architecture, metrics and exploration algorithm. The paper discusses also an application of the proposed exploration technique based on the Pareto Simulated Annealing algorithm, comparing the results with the full search exploration. For the selected set of benchmarks, the PSA techniques is up to three orders of magnitude faster than the full search, while maintaining its accuracy within 4% on average.

References

1. C. L. Su and A. M. Despain. Cache design trade-offs for power and performance optimization: A case study. In *ISLPED-95: ACM/IEEE Int. Symposium on Low Power Electronics and Design*, 1995.
2. Y. Li and J. Henkel. A framework for estimating and minimizing energy dissipation of embedded hw/sw systems. In *DAC-35: ACM/IEEE Design Automation Conference*, June 1998.
3. J. K. Kin, M. Gupta, and W. H. Mangione-Smith. Filtering Memory References to Increase Energy Efficiency. *IEEE Trans. on Computers*, 49(1), Jan. 2000.
4. T. M. Conte, K. N. Menezes, S. W. Sathaye, and M. C. Toburen. System-level power consumption modeling and tradeoff analysis techniques for superscalar processor design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 8(2):129–137, Apr. 2000.
5. Doug Burger, Todd M. Austin, and Steve Bennett. Evaluating future microprocessors: The simplescalar tool set. Technical Report CS-TR-1996-1308, University of Wisconsin, 1996.
6. David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings ISCA 2000*, pages 83–94, 2000.
7. N. Bellas, I. N. Hajj, D. Polychronopoulos, and G. Stamoulis. Architectural and compiler techniques for energy reduction in high-performance microprocessors. *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, 8(3), June 2000.
8. N. Vijaykrishnan, M. Kandemir, M.J. Irwin, H.S. Kim, and W. Ye. Energy-driven integrated hardware-software optimizations using simplepower. In *ISCA 2000: 2000 International Symposium on Computer Architecture*, Vancouver BC, Canada, June 2000.
9. Tony D. Givargis and Frank Vahid. Platune: a tuning framework for system-on-a-chip platforms. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21(11):1317–1327, November 2002.
10. M. Palesi and T Givargis. Multi-objective design space exploration using genetic algorithms. In *Proceedings of the Tenth International Symposium on Hardware/Software Codesign, 2002. CODES 2002*, May 6–8 2002.
11. D. Gajski, N. Dutt, A. Wu, and S. Lin. *High-Level Synthesis, Introduction to Chip and System Design*. Kluwer Academic Publishers, 1994.
12. K. Keutzer, S. Malik, A. R. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli. System level design: Orthogonalization of concerns and platform-based design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12):1523–1543, December 2000.
13. A. Aho, J. Hopcroft, and J. Ullman. *Data Structures and Algorithms*. Addison-Wesley, Reading, MA, USA, 1983.
14. Anatoly A. Zhigljavsky. *Theory of global random search*, volume 65. Kluwer Academic Publishers Group, Dordrecht, 1991.
15. Jaszkievicz A. Czyzak P. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimisation. *Journal of Multi-Criteria Decision Analysis*, (7):34–47, April 1998.
16. R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–140, 1994.