A Variability-Aware Robust Design Space Exploration Methodology for On-Chip Multiprocessors Subject to Application-Specific Constraints

GIANLUCA PALERMO, CRISTINA SILVANO, and VITTORIO ZACCARIA,
Politecnico di Milano

Manufacturing process variation is dramatically becoming one of the most important challenges related to power and performance optimization for sub-90nm CMOS technologies. Process variability impacts the optimization of the target system metrics, that is, performance and energy consumption by introducing fluctuations and unpredictability. Besides, it impacts the parametric yield of the chip with respect to application level constraints by reducing the number of devices working within normal operating conditions.

The impact of variability on systems with stringent application-specific requirements (such as portable multimedia and critical embedded systems) is much greater than on general-purpose systems given the emphasis on predictability and reduced operating margins. In this market segment, failing to address such a problem within the early design stages of the chip may lead to missing market deadlines and suffering greater economic losses.

In the context of a design space exploration framework for supporting the platform-based design approach, we address the problem of robustness with respect to manufacturing process variations. First, we apply Response Surface Modeling (RSM) techniques to enable an efficient evaluation of the statistical measures of execution time and energy consumption for each system configuration. Then, we apply a robust design space exploration framework to afford the problem of the impact of manufacturing process variations onto the system-level metrics and consequently onto the application-level constraints. We finally provide a comparison of our design space exploration technique with conventional approaches on two different case studies.

Categories and Subject Descriptors: C.0 [General]: System architectures; C.1.4 [Processor Architectures]: Parallel Architectures; C.3 [Special Purpose and Application-Based Systems]: Real-time and Embedded Systems, Microprocessor/Microcomputer Applications; C.4 [Performance of Systems]: Design Studies, Modeling Techniques

General Terms: Design, Performance

Additional Key Words and Phrases: Design space exploration, performance estimation, multiprocessors, multiobjective optimization, design of experiments, response surface modeling

ACM Reference Format:
DOI = 10.1145/2220336.2220341 http://doi.acm.org/10.1145/2220336.2220341

This work was supported in part by the EC under grant MULTICUBE FP7-216693.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from the Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, fax +1 (212) 869-0481, or permissions@acm.org

© 2012 ACM 1539-9087/2012/07-ART29 $15.00
DOI 10.1145/2220336.2220341 http://doi.acm.org/10.1145/2220336.2220341

1. INTRODUCTION

Nowadays, Chip Multiprocessors (CMP) are becoming an attractive alternative for application specific processor development. In fact, they represent the best compromise in terms of a stable hardware platform which is software programmable, thus customizable, upgradable and extensible. In this sense, the CMP paradigm minimizes the risk of missing the time-to-market deadlines while ensuring greater efficiency due to memory subsystem customization and software compilation techniques.

As a result, we increasingly use general purpose homogeneous computing techniques in the application specific scenario (such as, for example, GPGPUs, software defined radios [Bougard et al. 2008], multimedia acceleration [De Sutter et al. 2006]), supported by advanced parallel programming techniques (such as stream programming) to deal with application-specific tasks [Gordon et al. 2002, 2006].

IP reuse, platform reconfigurability and homogeneous programmability approaches are pushing towards a new design paradigm [Keutzer et al. 2000], which is strongly influencing today’s automatic synthesis of embedded systems. In this context, a virtual microprocessor-based architecture can be easily extended and customized for a particular application, enabling a quick, low-risk deployment. More specifically, preverified components belonging to a specific library are instantiated and sized to meet specific constraints on the target application domain. However, the space of configurations (or “design space”) of each of the system components can be very large. Considering a CMP, a reasonable set of architectural parameters can be the number of processor cores, the number of parallel issues per core, the number of levels in the memory hierarchy and positioning (on-chip, off-chip), cache configuration parameters (cache size, block size, associativity, unified vs split data/instruction caches, at any hierarchy level, etc.), bus topologies and channel width.

Process variation is critically becoming one of the most significant challenges related to power and performance optimization for sub-90 nm CMOS technologies. Parametric yield, that is, the percentage of dies that meet power and performance constraints, has become as important as power and performance optimization itself.

Manufacturing process variability is mainly due to interdie and intradie variations. Inter- and intradie variations affect low level process parameters such as the channel gate length, the thickness of the oxide and the threshold voltage, which, in turn, affect the critical path delay and static and dynamic power consumption. Interdie fluctuations affect uniformly every element on a die and consist of lot-to-lot and wafer-to-wafer variations such as processing temperatures, equipment properties, wafer polishing, wafer placement and the resist thickness. Conversely, intradie parameter fluctuations consist of both random and systematic components and generate nonuniform electrical characteristics across the chip [Bowman et al. 2002].

Estimating the impact of parameter fluctuations on circuit performance (i.e., Variability-Aware Modeling) is of extreme importance for maximizing the company’s overall revenue. Generally, overestimation impacts the design complexity and the design time with consequences on the die size and the time-to-market window. On the other hand, underestimation can impact the product performance and yield as well as increase the silicon debug time. Overall, overestimation impacts the design effort while underestimation impacts the manufacturing effort.

The impact of variability on systems with stringent application-specific requirements such as portable multimedia and critical embedded systems is much greater than on general-purpose systems given the emphasis on predictability and reduced operating margins to guarantee a specific level of quality-of-service (QoS). In this market segment, failing to address such a problem within the early design stages of the chip may lead to missing market deadlines and suffering greater economic losses.
In this scenario, we address the problem of variability-aware design at system-level for CMP, focusing on the impact of manufacturing process variations onto the system-level metrics and consequently onto the application-level constraints.

This article tackles the problem of CMP robust multiobjective optimization by extending and integrating, into a case study, our previous work [Palermo et al. 2009a, 2009b] to address process manufacturing variability. In particular, we use a clever formulation of the objectives which is obtained by aggregating variability data as Taguchi’s outer arrays to build appropriate Taguchi’s signal-to-noise ratios combined with response surface modeling to prune actual simulations from the core optimization loop. Although, in principle, some of the used techniques are not new [Jin and Branke 2005; Palermo et al. 2009a; Tsai et al. 2004], this article presents a full-fledged implementation of a robust design space exploration framework to tackle the problem of the impact of manufacturing process variations at system-level. The proposed approach is supported by an extensive validation and experimentation strategy targeted to an MPEG video decoder and an embedded PC scenario.

More in detail, the problem of process variability is addressed by means of an accurate tuning of the system-level parameters by applying the following.

— A robust design space exploration (DSE) framework. The main goal of this framework is the tuning of the target architecture parameters towards the minimization of the variance of the system metrics (e.g., QoS or performance) and the maximization of the overall parametric yield (considering both die-to-die and within-die process variation impact on the application specific constraints). In this view, the article is a step forward with respect to conventional approaches [Ascia et al. 2007; Palermo et al. 2008a, 2008b] while being orthogonal to low-level circuit optimizations or dynamic corrections such as dynamic compensation [Sanz et al. 2006].

— Response surface modeling. We speed up the exploration process by using response surface modeling (RSM) techniques to tackle the additional complexity due to the elaboration of variability effects. RSMs will be used selectively instead of real simulations for the estimation of the system-level metrics associated to each system configuration, reducing the overall exploration effort.

The DSE framework is based on a set of state-of-the-art accurate performance, area and energy models of a CMP taking into account process variations at the 70nm technology node (see Table I). Although the proposed methodology is general enough to be easily retargetable to other CMP architectures and technology nodes, in this article we target a MIPS multiprocessor architecture. To estimate system-level metrics, we leveraged SESC [Renau et al. 2005] simulation tool which represents a state-of-the-art MIPS instruction set simulator for CMPs providing dynamic energy and execution cycles. We have also extended the performance and power models with an area estimation model inspired by Kalla et al. [2004] and Li et al. [2006].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td># Processors</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Processor issue width</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>L1 instruction cache size</td>
<td>2K</td>
<td>16K</td>
</tr>
<tr>
<td>L1 data cache size</td>
<td>2K</td>
<td>16K</td>
</tr>
<tr>
<td>L2 private cache size</td>
<td>32K</td>
<td>256K</td>
</tr>
<tr>
<td>L1 instruction cache assoc.</td>
<td>1w</td>
<td>4w</td>
</tr>
<tr>
<td>L1 data cache assoc.</td>
<td>2w</td>
<td>8w</td>
</tr>
<tr>
<td>L2 private cache assoc.</td>
<td>2w</td>
<td>8w</td>
</tr>
<tr>
<td>I/D/L2 block size</td>
<td>16</td>
<td>32</td>
</tr>
</tbody>
</table>

Table I. Design Space for the Target CMP Platform
The proposed methodology has been tuned and validated by using the SPLASH-2 [Woo et al. 1995] parallel benchmark suite while the efficacy and efficiency has been analyzed with two use cases.

— **Portable MPEG decoder.** This use case is a characteristic example of a modern portable multimedia system, where the QoS constraints can be directly inferred in terms of video frame-rate.

— **Industrial embedded PC.** This use case considers industrial embedded PCs used for analysis and control. These systems can run complex and differentiated workloads on top of general purpose OSes like Windows XP. To tackle this scenario, we consider a system with a wider target set of target applications characterized by tight constraints on power density and consumption such as those imposed by an industrial embedded deployment scenario.

The article is organized as follows. While Section 2 gives an overview of the related work on variability-aware modeling at architectural level and on design space exploration techniques, Section 3 briefly introduces the background on reference, state-of-the-art performance, and power models used for estimating the impact of process variations on the system. Section 4 introduces the response surface model for the dynamic power and execution cycles we used while Section 5 introduces the design space exploration methodology proposed in this article. Section 6 shows the experimental results of the proposed methodology and finally, Section 7 highlights the article conclusions.

2. RELATED WORKS

In recent years, the problem of the process variability impact on the performance and leakage power consumption started to gain relevance and attention [Eisele et al. 1997]. Bowman et al. [2002] have proposed statistical models for both within-die and die-to-die variations, investigating the correlation with microarchitecture-driven parameters such as the number of critical paths. Recently Bowman et al. [2007], have extended the analysis of the variation on the critical path to the multiprocessor domain. Borkar et al. [2003], show that the choices made at the micro-architecture level affect the variability of the overall chip, advocating probabilistic optimization to optimize clock speed, energy consumption and total area. The authors propose the use of adaptive body biasing in order to decrease the effect on parameter variations on the speed and leakage power. Marculescu and Talpes [2005], propose microarchitecture-level models for within-die process variability to be included in the design of high-performance processors. The authors perform also a limited design space exploration of fully synchronous and globally-asynchronous-locally-synchronous systems by taking into account process variability. Grabner et al. [2006], propose a technique to estimate the system level parametric yield loss for a set of alternative memory configurations. The approach aids the designer to make educated trade-offs between power consumption and timing yield.

Regarding the design space exploration problem, several methods have been recently proposed in literature. Those techniques can be classified in two main categories: heuristics for architectural exploration [Ascia et al. 2007; Palermo et al. 2006] and methods for the system performance estimation and optimization [Ipek et al. 2006; Joseph et al. 2006b; Lee and Brooks 2006; Palermo et al. 2008a].

Palermo et al. [2006], compare Pareto Simulated Annealing, Pareto Reactive Taboo Search, and Random Search exploration to identify energy-performance trade-offs for a parametric super-scalar architecture executing a set of multimedia kernels. In the
same direction but more complex is the combined Genetic-Fuzzy system approach proposed in Ascia et al. [2007]. The technique is applied to a highly parametrized SoC platform based on a VLIW processor in order to optimize both power dissipation and execution time. The technique is based on a Strength Pareto Evolutionary Algorithm coupled with fuzzy system rules in order to speedup the evaluation of the system configurations.

State-of-the-art for the system performance optimization is presented in [Ipek et al. 2006; Joseph et al. 2006b; Lee and Brooks 2006; Palermo et al. 2008a]. A common trend among those methods is the combined use of response surface modeling and design of experiments methodologies. In Joseph et al. [2006b], a Radial Basis Function has been used to estimate the performance of a super-scalar architecture; the approach is critically coupled with an initial training sample set that is representative of the whole design space, in order to obtain a good estimation accuracy. The authors propose to use a variant of the Latin Hypercube method in order to derive an optimal, initial set. In Lee and Brooks [2006] and Palermo et al. [2008a] linear regression has been used for the performance prediction and assessment. The authors analyze the main effects and the interaction effects among the processor architectural parameters. In both cases, random sampling has been used to derive an initial set of points to train the linear model. A different approach is proposed in Ipek et al. [2006], where the authors tackle performance prediction by using an Artificial Neural Network paradigm to estimate the system performance of a CMP.

Finally, Gerstlauer et al. [2009] and Nikolov et al. [2008] design space exploration framework for heterogeneous MPSoCs which exploits high-level modeling based on Kahn Process Networks. The framework is capable to perform hardware/software partitioning and synthesis on multiprocessor systems with hardware accelerators.

Concerning Robust Optimization, Beyer and Sendhoff [2007] and Jin and Branke [2005] present a comprehensive survey of state of the art approaches concerning uncertainty (or noise) on the objective functions and the design parameters. The authors, provide a comprehensive framework to categorize the current approaches to multiobjective robust optimization. Our article builds upon the directions indicated by the authors by both addressing multi-objective optimization and response surface modeling into a unique framework.

Among the most important papers on robust multiobjective optimization we can find Deb and Gupta [2006]; in that paper, instead of optimizing the original objective functions, the authors optimize the mean effective objective functions computed by averaging a representative set of neighboring solutions (in the parameter space) and introduce a new definition of robustness by optimizing the original objectives but adding a constraint limiting the extent of functional change by local perturbations to a user-defined value. Our work differs from this approach due to the fact that the perturbations considered here are not associated with the parameter space but only with the objective space.

Jin and Branke [2005], treat a robust single-objective problem explicitly as a multiobjective optimization task, identifying the trade-off between nominal value and variance (robustness) in the form of the obtained Pareto front. As will be seen in the rest of the article, our approach differs from the above in the sense that it tries to minimize the number of objectives of the optimization problem. Finally, Lim et al. [2006] define an inverse robust approach, where starting from the desired performance, the algorithm searches for solutions that guarantee a certain degree of maximum uncertainty and at the same time satisfy the desired nominal performance of the final design solution. For this purpose, a series of nested multipoint local searches is conducted to address uncertainty on design variables.
In this article, we specifically target a homogeneous microprocessor-based architecture which can be easily extended and customized for a particular application, enabling a quick, low-risk deployment. In particular, we assume to have a set of preverified microprocessor components which can be sized to meet specific constraints on the target application domain. The target architecture has been identified only as a workbench for demonstration purposes of the proposed methodology, which allows to reduce architectural optimizations to parameter optimization (cache sizes, processor number).

The demonstration CMP targeted in this article is composed of several out-of-order processor cores, containing private L1 and L2 caches (see Figure 1) where interprocessor communication leverages a high-bandwidth split-transaction bus supporting a MESI coherence protocol acting directly between L2 caches. The design space of the target CMP architecture we address in this article is composed of the parameters listed in Table I, where the minimum and maximum values have been reported.

In the following analysis, we target a 70 nm technology node but the techniques that we are presenting can be applied to future smaller technology nodes. The prediction of the energy and time computed by the model is probabilistic in the sense that it is modeled by a specific probability distribution function (PDF). The model we use for energy and delay variability is derived from the state of the art literature.

The reference execution time model of a specific application depends essentially on the number of cycles per application execution and the maximum critical path delay. The maximum critical path delay can be assumed (as suggested in Bowman et al. [2002] and Srivastava et al. [2005]) as a random variable with normal distribution and which is inversely correlated with the leakage power (described later).

As projected by the 2003 ITRS road-map for processing units (or MPUs), we assume an asymptotic critical path delay of 12 FO4 inverters for each technology node. Table II shows the reference nominal values for the critical path delay [Hu et al. 2000] as well as the number of projected critical paths (i.e., the number of paths contributing to the total critical path delay) for each technology node [Bowman et al. 2002].
Considering process variations, the actual critical path delay of the target processor becomes a random variable whose distribution is given by the nominal delay and intradie and interdie components [Bowman et al. 2002]. The total probability distribution of the critical path delay $D$ is the following:

$$f(D) = \delta(D - D_{nom}) * g_n(D) * h(D),$$  \hspace{1cm} (1)

where $\delta$ is the convolution operator, $\delta(D_{nom})$ is an impulse at $D_{nom}$, $g_n(D)$ is the delay distribution due to intradie process variations for $n$ critical paths contributing to the total delay, and $h(D)$ is the delay distribution due to interdie process variations.

Previous work [Bowman et al. 2002; Marculescu and Tulpes 2005] proposed some approximations for Equation (1). Based on [Bowman et al. 2002], we can assume that, for a high number of critical paths ($n \geq 1000$), the distribution of $g_n$ is an impulse and determines only the mean of the distribution $f(D)$. For a 70 nm technology node, an average 26% increase on $D_{nom}$ can be expected for $10^4$ critical paths. The distribution of $h(D)$ is a normal distribution with mean 0 and variance $\sigma_h^2$. According to Bowman et al. [2002], Nassif et al. [2007], and Srivastava et al. [2005] we assume a worst-case value of $\sigma_h/D_{nom} = 9\%$.

The average energy dissipation $E$ of a CMP processor depends on the actual number of cycles of execution, the critical path delay and the power consumption $P$, which can be roughly decomposed in two components:

$$P = P_{dynamic} + P_{leak}.$$  \hspace{1cm} (2)

The term $P_{dynamic}$ is the power consumption of the system due to switching activity of the internal nodes and the capacity driven by those nodes. Based on [Borkar et al. 2003; Rao et al. 2005; Srivastava et al. 2005], we can assume that $P_{dynamic}$ varies with the maximum allowable frequency and it is dependent on the total switching capacitance $\sum_i C_i$ for the target technology node.

$$P_{dynamic} = \frac{1}{2} \alpha \sum_i C_i \frac{V^2_{dd}}{D}.$$  \hspace{1cm} (3)

Obviously, the dynamic power is dependent on the given application and on the configuration of the target architecture. This contribution is usually estimated by using simulation techniques. Also in this case we will leverage the CMP simulation tool [Renau et al. 2005] to estimate dynamic energy of our application. Among other simulation techniques, our simulation tool is based on the CACTI [Wilton and Jouppi 1996] tool for computing the energy consumption of the caches.

The leakage power $P_{leak}$ is the power due to static leakage currents [Rao et al. 2005]. Following Srivastava et al. [2005] and Rao et al. [2005] we can assume that the leakage power has a log-normal distribution. In other words, the logarithm of the power consumption is a normal (Gaussian) random variable; If the power consumption has a mean $\mu_{leak}$ and variance $\sigma_{leak}^2$, the logarithm is a normal variable with the following mean and variance:

$$\mu_{log.leak} = \ln(\mu_{leak}) - \frac{1}{2} \ln \left(1 + \frac{\sigma_{leak}^2}{\mu_{leak}}\right)$$  \hspace{1cm} (4)

$$\sigma_{log.leak}^2 = \ln \left(1 + \frac{\sigma_{leak}^2}{\mu_{leak}}\right)$$  \hspace{1cm} (5)

According to Srivastava et al. [2005], Nowak [2002], and O’Connor [2006], we can use a projected value of leakage power density around (3W/cm²) for a 70nm technology node.
Although, as the area of a module increases, random variations tend to decrease the ratio between standard deviation and the average power, we currently lack information on the specific behavior. For this reason we consider a conservative upper-bound $\sigma_{\text{leak}}/\mu_{\text{leak}}$ ratio of 20% projected from data in Srivastava et al. [2005].

As a matter of fact, both $D$ and $\ln(P_{\text{leak}})$ terms follow a bivariate Gaussian distribution and they are inversely correlated by a correlation factor $\rho$. Although the actual layout of the processor is really important for determining the coefficient of correlation, we actually lack specific configuration-dependent information for $\rho$. Nevertheless, in the current literature, typical values of $\rho$ range from $-0.65$ to $-0.95$ (average $-0.87$), as reported in Srivastava et al. [2005]. We will assume an average value of $\rho = -0.87$.

4. EFFICIENT EVALUATION OF THE SYSTEM METRICS BY USING RESPONSE SURFACE MODELING TECHNIQUES

In this article, we propose an efficient design space exploration strategy leveraging randomized design of experiments (DoE) and linear response surface modeling (RSM) techniques, in the spirit of Palermo et al. [2009a]. Given the objective functions associated to the system, the methodology enables the efficient identification of an approximate Pareto set of candidate architectures by minimizing the number of simulations of system configurations. This is a notable achievement, since, nowadays, evaluating the objective function $\hat{f}(\vec{x})$ of a single system configuration $\vec{x}$ (being it either performance or power consumption) means hours or days of simulations under a realistic workload for complex SoCs.

The term randomized Design of Experiments (DoE) [Santner et al. 2003] is used to identify the off-line planning of a simulation campaign where a set of variable architectural parameters are randomly tuned. On the other hand, Response Surface Modeling techniques are used to define an analytical dependence between several design parameters and one or more response variables. The working principle of RSM is to use a set of simulations generated by DoE in order to obtain a response model of the system. A typical RSM-based flow involves a training phase, in which known data (or training set) is used to identify the RSM configuration, and a prediction phase in which the RSM is used to forecast an unknown system response. RSMs are an effective tool for analytically predicting the behavior of the system platform without resorting to a system simulation. RSM-based techniques represent the kernel of the methodology. In this article, we use a linear regression technique for creating an analytic model of the system metrics associated with the target architecture [Joseph et al. 2006a; Lee and Brooks 2006].

4.1 Linear Regression Theory

Linear regression is a well-known regression method that models a linear relationship between a dependent scalar figure of merit $y$ and some independent variables $x_i (i = 1 \cdots p)$ plus a random term $\varepsilon$. Such type of model contains both simple as well as exponential terms combined linearly. However, research efforts [Myers et al. 1989] have mainly been addressed to first and second order models. The general expression for a second order linear surface model is given by:

$$y(\vec{x}) = \hat{y}(\vec{x}) + \varepsilon = \alpha_0 + \sum_{k=1}^{p} \alpha_k x_k^2 + \sum_{i=1}^{p} \sum_{k=1,k\neq i}^{p} \beta_{i,k} x_i x_k + \sum_{k=1}^{p} \gamma_k x_k + \varepsilon,$$

where $\hat{y}(\vec{x})$ is called a prediction of system metric $y$. In this section, we assume to work with a single scalar metric $y$; however, in the rest of the article we will work with a set of different models, each associated with the different target system metrics.
In the following part of the section we will identify with $y_j$ the actual value of the metric $y$ derived with the $j$-th simulation, while $\hat{y}_j$ will be the associated prediction. Least squares analysis can be used to determine a suitable estimate for the coefficients $\alpha, \beta, \gamma$ in order to minimize the value of the squared sum of residuals:

$$\sum_{j=1}^{S} (\hat{y}_j - y_j)^2$$

over a set of $S$ simulations used to build the model in Equation (6).

A measure of the quality of fit associated with the resulting model is called coefficient of determination:

$$R^2 = \frac{SSR}{SST}$$

where

$$SST = \sum_j (y_j - \bar{y})^2$$

is the total sum of squares of simulations $y_j$ and the average simulation $\bar{y}$, and

$$SSR = \sum_j (\hat{y}_j - \bar{y})^2$$

is the regression sum of squares between the prediction for simulation $j$ (i.e., $\hat{y}_j$) and the average of simulations $\bar{y}$. As a rule of thumb, the higher $R^2$, the better the model fits the data. A value of $R^2$ equal to 1.0 indicates that the regression line perfectly fits the data. In this article, we will use the definition of the adjusted $R^2$. This is a modification of $R^2$ that adjusts for the number of explanatory terms in a model. The adjusted $R^2$ increases only if the terms of the model improve it more than expected by chance and will always be less than or equal to $R^2$; it is defined as:

$$1 - (1 - R^2) \frac{n-1}{n-p-1},$$

where $p$ is the total number of terms in the linear model, not counting the constant term, while $n$ is sample size. Adjusted $R^2$ is particularly useful in the feature selection stage of model building.

4.2 Linear Regression Model Selection

In order to understand the optimal number of terms of the linear model and the corresponding model order, we analyzed the behavior of the RSM cross-validation error and adjusted $R^2$ by varying the number of random training samples (derived from the simulations of the target applications). The cross-validation consisted of predicting both energy consumption and execution cycles for a set of 15,000 random system configurations (among the 55,296 of the whole design space presented in Table I) not used during the training of the model. The training samples for the model have been varied from 200 (lower bound given by the total amount of parameters) and 1600, the latter being almost 10% of the validation set.

Equation (11) represents an improved measure of the overall quality of fit of the linear regression: it is inversely proportional to the model’s degrees of freedom (i.e., $n - p - 1$) which, in turn, depend on the order of the chosen polynomial $f(\vec{x})$. As a matter of fact, higher degrees of freedom increase the chance of reduced variance of the model coefficients thus improving model stability while avoiding overfitting.
Our heuristic model selection tries to maximize the number of degrees of freedom and, at the same time, to minimize (in the order of 200) the number of simulations needed to build the model in the initial iterations of the methodology. Thus, as a “rule of thumb,” we decided to set the maximum number of terms to 50 to increase the chance of good quality of fit. Eventually, we limited the set of considered models to the following configurations:

1. first-order model, without any interaction between parameters (9 terms);
2. first-order model, with interaction between parameters \(9 + \binom{9}{2} = 45\) terms included parameter cross-products;
3. second-order model, without any interaction between parameters (18 terms).

The main problem associated with RSMs is that the prediction accuracy can be, to some extent, negatively impacted by the variance of the observation data. Thus, to further improve the prediction accuracy, we introduce a Box-Cox power transform on each sample \(y\) of the observed data [Joseph et al. 2006a]. The Box-Cox transform is a useful data (pre)processing technique used to reduce data variation, make the data more normal distribution-like and improve the correlation between variables. The power transformation is defined as a continuously varying function, with respect to the power parameter \(\lambda\):

\[
y^{(\lambda)} = \begin{cases} 
(y^\lambda - 1)/\lambda, & \text{if } \lambda \neq 0 \\
\log y, & \text{if } \lambda = 0
\end{cases}
\]  

In this article, we considered \(\lambda \in \{1, 0.5, 0, -1\}\). This range is inspired by the range originally considered in the Box and Cox [1964], where the power transformation was first proposed. For each transformation, we selected a set of random configurations as input to the linear regression model and computed the maximum normalized error on the training set. Besides, we remark that an inverse Box-Cox transformation has been applied on the model predicted data to keep it consistent with the target problem objective functions.

The following model configurations have been tested:

- first order model, without interaction between parameters;
- first order model, with interaction between parameters;
- second order model, without interaction between parameters.

We note that adding predictors to the linear model increases the chances of reducing the prediction error. However this is at the expense of an increased size in the model training set and increased complexity in the predictors computation (and associated numerical stability). Moreover it does not guarantee an increase on the adjusted \(R^2\). We thus limited ourselves to the second order model without interaction.

To perform the model calibration for the target CMP architecture, we used applications derived from the Stanford Parallel Applications for Shared Memory (SPLASH) [Woo et al. 1995] suite to gather information about the execution cycles and dynamic power consumption. The SPLASH-2 suite is organized into two sections: kernels and applications. We selected a partial subset of the suite as follows.

- Kernels. Complex 1D FFT (\texttt{fft}), Blocked LU Decomposition (\texttt{lu}), Blocked Sparse Cholesky Factorization (\texttt{cholesky}), Integer Radix Sort (\texttt{radix}).
Figure 2 shows the geometric average of the maximum RSM prediction error over the set of selected applications (left side of Figure 2) and the average adjusted $R^2$ (right side of Figure 2) for the three different model configurations: first order without interactions (2(a)), first order with interactions (2(b)) and second order without interactions (2(c)). Overall, we note that the best Box-Cox output transformation is the $\lambda = 0$ function. The prediction error associated with this transform in the second order model configuration is almost half the one associated with the other two model configurations. Moreover, Figure 2(c) shows that, for the same Box-Cox transformation, the adjusted $R^2$ is between 4% and 6% higher that the other two model configurations.

It is worth noting that increasing the number of training samples may have benefits, from a coarse-grained point of view, for both the normalized error and the coefficient of determination. However, as we deal with a random selection of training samples, there may be situations in which the new samples produce a fluctuation on the model coefficients. On one hand, to some very little extent, this can decrease the overall...
model accuracy either due to an overfitting on the training set or an increase in the training data variance due to an unstable Box-Cox transform such as $\lambda = -1$.

For the sake of clarity, we felt that 1600 samples was a suitable upper limit to be considered for all the benchmarks given the $\lambda = 0$ transform (which is the one who better stabilized the variance of the training set and, thus, the error). These conclusions are generalizable with care given that they depend on the size of the design space and the variability of the workloads.

As the second order model shows the best behavior in terms of average prediction error and adjusted $R^2$, we analyze more in detail the behavior of the RSM for the Application set of the SPLASH-2 benchmark suite (barnes, ffm, ocean and volrend) separating the analysis between the execution cycles and dynamic power predictions.

Figure 3 shows the RSM properties when predicting the execution cycles of each single application in terms of maximum prediction error and coefficient of determination $R^2$. In particular, Figure 3(b) and 3(d) show that the $fmm$ and $volrend$ applications are very predictable with a very small maximum prediction error. In the other cases (as in Figure 3(a) and 3(c)), the maximum prediction error prediction for $barnes$ and $ocean$ application is very dependent on the Box-Cox transformation used. Finally, as confirmed by the right part of Figure 3, the logarithmic Box-Cox transform ($\lambda = 0$) is the one which produces the highest value for $R^2$.

Figure 4 shows the RSM properties when predicting the energy consumption of each single application in terms of maximum prediction error and coefficient of determination adjusted $R^2$. The figure underlines that the Box-Cox transform is very important also in predicting the energy consumption. In the worst-case ($fmm$ application, Figure 4(b)) it allows us to reduce by up to one order of magnitude the maximum prediction error of the plain response function ($\lambda = 1$); the prediction error reduces of up to four orders of magnitude with respect to the case $\lambda = -1$. Also in these cases, the logarithmic Box-Cox transform ($\lambda = 0$) is the one showing the overall lowest error.

It is interesting to note that in both cases the Box-Cox transform has a very high influence on the adjusted $R^2$. Overall the adjusted $R^2$ can increase up to 30% if considering $\lambda \neq 1$. The final results of the model selection indicate that the second-order linear model with Box-Cox $\lambda = 0$ (meaning a $\log(y)$ transformation) is the best model for predicting both the energy and execution cycles of our CMP applications.

5. A ROBUST DESIGN SPACE EXPLORATION METHODOLOGY

In this section we exploit the techniques presented in Sections 3 and 4 to create a full fledged implementation of a robust design space exploration framework to tackle the problem of the of manufacturing process variability.

In particular, our target problem is to identify a set of configurations of the system that minimize the variance over a number of manufactured chips with respect to a system level figure of merit $y$, while optimizing the nominal (or mean) value of $y$. Overall, we want to maximize the number of chips which are compliant with the application-level constraints by considering from a combined perspective both die-to-die and within-die process variations (i.e., parametric).

Figure 5 shows the part of the variability-aware design flow we used to obtain the evaluation of candidate configurations.

In our methodology, we define the system response $y$ as one of the aggregate system metrics derived from an actual simulation of the system (or estimated with an appropriated linear RSM, as described in Section 4.1) and its variability-dependent components (the probabilistic models, as defined in Section 3).

Given the nonlinear, probabilistic nature of the system, it is very difficult to find closed form solutions to our problem. We thus resort to a sampling technique of the
Fig. 3. RSM maximum error (left) and coefficient of determination adjusted $R^2$ (right) for the execution cycles metric. Second order linear model, no interaction.
Fig. 4. RSM maximum error (left) and coefficient of determination adjusted $R^2$ (right) for the energy consumption metric. Second order linear model, no interaction.
distribution probability (both for the power consumption and frequency) to derive a unique measure of the nominal performance of the system as well as its variance.

For a specific system level metric $y$ and a set of $N$ samples $y_i$ picked up from the random variable $y$, we define the following statistical measure:

$$Q_y = \frac{1}{\frac{1}{N} \sum_{i=1}^{N} y_i^2}.$$  
(13)

This function is called quality measure and increases monotonically as long as the variance of the metric $y$ and its nominal value decrease. In other words, it is an aggregate measure of both mean and variance [Song et al. 1995; Taguchi 1987] which is very suitable to be used whenever the metric and its variance should be minimized (also called as-small-as-possible metric). A dual definition can be introduced whenever the system metric should be maximized but the variance should be minimized (also called as-large-as-possible metric [Taguchi 1987]).

The quality function is derived from the Taguchi’s signal-to-noise ratio. The idea behind this kind of measures is to treat the variability as random noise with respect to which the nominal value of metric should be optimized. The higher the signal-to-noise ratio, the lower is the variance compared to the expected nominal value of the metric. The quality measures are tools for managing the probability distributions associated with process variability. Essentially, we transform a random-variable (the system response to be optimized) into an aggregate sampled measure which will be used as an optimization objective in the following steps of our methodology. The introduction of these quality metrics enables us to propose a new DSE methodology which, compared to conventional approaches, is more efficient and flexible. Experimental evidence to this statement will be provided in the experimental results section.

Following the previous definitions, the Pareto solutions of the variability-aware design space exploration problem that we are facing with is the set of configurations which are optimal from the point of view of the quality metrics and parametric yield:

$$\max_{\forall \bar{x} \in \mathbb{X}} \begin{bmatrix} Q_1(\bar{x}) \\ \vdots \\ Q_m(\bar{x}) \\ Y(\bar{x}) \end{bmatrix},$$  
(14)

where $Q_k(\bar{x})$ is the quality metric $k$ associated to configuration $\bar{x}$ while $Y(\bar{x})$ is the overall parametric yield on the application requirements.

As mentioned in the previous section, the multiobjective optimization meta-heuristic used in this article, to find the Pareto solutions of the problem stated in Equation (14), is inspired by [Palermo et al. 2009a] and leverages a random design of experiments technique coupled with a linear regression RSM. Recent studies [Palermo
et al. 2008a] confirmed that, in the field of multiprocessor DSE, a randomized design of experiments enables the best tuning of response surface models.

The multiobjective exploration algorithm is depicted in Figure 6 and can be described as follows.

1. Apply a randomized design of experiments plan to pick up the set of initial configurations set $S_0$. This step provides an initial coarse view of the target design space at iteration 0.

2. Run the simulations to obtain the dynamic power consumption and the execution cycles actual measurements associated to each configuration in $S_0$. Perform a Monte-Carlo sampling of the distribution probabilities associated to the overall execution time and overall energy (dynamic+leakage) and compute the quality measures associated to it.

3. Generate a linear regression RSM by using $S_0$ as training set. The RSM generates the new design space $\hat{S}_1$ composed of a set of estimated measurements for the cycles and the dynamic power associated to each configuration. Potentially, $\hat{S}_1$ could be as large as the entire design space, however a sampling technique could be used if it is not practically feasible to manage such a large design space. Perform a Monte Carlo sampling of the distribution probabilities associated to the overall execution time and overall energy (dynamic+leakage) and compute the quality measures associated to it.

4. Compute the Pareto front associated to the quality metrics for execution time and overall power: $\hat{P}_1 = Pareto(\hat{S}_1)$.

5. Run the simulations to derive the actual measurements on the architectural configurations contained in $P_1$. The result is the design space $P_1$.

6. If $P_1$ covers $P_0 = Pareto(S_0)$ by a percentage greater than zero and the stopping criterion is not met, restart from step 3, where now $S_0 \leftarrow S_0 + P_1$. The stopping criterion is the maximum number of actual measurements to be done.

To help the system architect to select among the large number of feasible solutions of the Pareto front, we can envision an approach that starts by clustering the architectural configurations and then selects a sort of “golden” solution for each cluster by

---

$^1$The formal definition of the coverage function $C(A, B)$ can be found in Okabe et al. [2003] and represents the percentage of points in $B$ that are strictly dominated by points in $A$. 

using a decision-making-mechanism. This approach is shown in the experimental results section.

6. EXPERIMENTAL RESULTS

In this section, the proposed methodology has been applied to the customization of two case studies based on the same CMP architecture: an MPEG2 decoder and an industrial embedded PC processor. The two case studies differ not only in terms of the target applications but also in terms of system-level constraints and objectives. The target architecture is the shared-memory CMP with private L2 cache previously presented in Section 3. We focused our analysis on the architectural parameters listed in Table I, where the minimum and maximum values have been reported. Globally, the resulting design space consists of 55,296 alternative configurations. In both cases, to carry out the nominal values of the system metrics, we leveraged the SESC [Renau et al. 2005] simulation tool, a fast simulator for CMP architectures with out-of-order processors that is able to provide energy and performance results for a given application. Within SESC, the energy consumption evaluation for the memory hierarchy is supported by CACTI [Wilton and Jouppi 1996], while the energy consumption evaluation due to the core logic is based on the WATTCH models [Brooks et al. 2000].

The SESC simulator has been chosen because it offers a good trade-off between simulation speed and modeling accuracy (less than 5% with respect to an actual implementation of the MIPS R10K architecture [Renau et al. 2005]) and it is widely adopted by the scientific community for CMP design space modeling and exploration [Ipek et al. 2008; Kim et al. 2008; Monchiero et al. 2008; Tiwari and Torrellas 2008; Teodorescu and Torrellas 2008].

For the analysis of the results obtained by the proposed methodology, we performed the following two steps for both case studies.

1. Application of the variability-aware robust DSE methodology. We applied the optimization algorithm presented in the previous section to the target CMP architecture. To help the system architect to select among all the feasible solutions in the Pareto front, we clustered the architectural configurations into 2 sets by using the k-means clustering algorithm. Then, we selected a “golden” solution for each cluster by using a decision-making-mechanism, analyzing the related performance;

2. Comparison with conventional approaches. To validate the proposed methodology, we compare it with respect to a conventional approach where the optimization problem has been formulated by imposing new sharp constraints on the application requirements (the constraints have been tightened by considering the expected standard deviation on the metrics) and by using the direct evaluation of the metrics with the nominal values. The comparisons have been done considering the same exploration effort in terms of time.

6.1 MPEG Decoder Case Study

In this section, the proposed methodology has been applied to the customization of a CMP architecture for the execution of an MPEG2 decoder application. In this case study, we used the ALPBench MPEG2decoder [Li et al. 2005]. The objective functions have been derived as a geometric average of the system metrics over a set of five input data-sets composed of 10 frames at a resolution equal to 640x480.

---

2A geometric mean tends to dampen the effect of very high or low values, which might bias the mean if a straight average (arithmetic mean) were calculated [Palermo et al. 2008b].
For this application-specific customization, the multiobjective optimization problem has been formalized as follows:

\[
\max_{\vec{x} \in X} \left[ \frac{1}{Q_{\text{energy, per frame}}(\vec{x})} \right],
\]

where the parametric yield has been defined as:

\[
Y(\vec{x}) = \text{prob} \{ \text{frame rate}(\vec{x}) \geq 25, P_{\text{dens}} \leq 60 \text{W/cm}^2, P \leq 25 \text{W} \}
\]

and subject to the following constraints:

\[
\text{total system area}(\vec{x}) \leq 85 \text{mm}^2.
\]

The optimization problem has four objective functions which are the total system area, the energy consumption per frame, the frame rate and the yield of the target system with respect to a set of application requirements. The \(Q_{\text{energy, per frame}}(\vec{x})\) is a as-small-as-possible quality measure while \(Q_{\text{frame rate}}(\vec{x})\) is a as-large-as-possible quality measure. The application requirements are as follows.

— **The minimum frame rate.** This is introduced as a Quality of Service (QoS) constraint considering a standard 50 half-frame per second.

— **The average power density.** Following ITRS [Kahng 2002] we consider an upper bound of 60 W/cm\(^2\) for the maximum power dissipation of cost-effective packaging.

— **The average power dissipation.** This constraint is given in terms of desired battery lifetime.

The area constraint (Equation (17)) has been defined to impose an overall upper bound to the cost of manufacturing and packaging.

The decisions we made in determining the above objectives and yield are indeed not unique and are subject to specific considerations associated with the target use-case. We must bear in mind that adding objectives to the optimization problem is likely to increase the number of points in the Pareto-front, thus increasing the complexity of the final decision making. Having said that, not all figures of merit are interesting to be traded-off since, as in the previous case study, we need only to maximize the percentage of chips that meet the constraint(s). In particular, the power and power density have been aggregated into the yield (see Equation (16)), since we are not interested in any trade-off between these two figure of merit.

**Application of the variability-aware robust DSE methodology.** The randomized design of experiments generated as starting point for the optimization methodology is composed of a set of 250 simulations, while an additional number of 70 simulations has been generated by the optimization algorithm. Globally, only 5.8‰ of the target design space has been simulated. The final approximated Pareto front is composed of 11 configurations.

To help the system architect to select among the feasible solutions in the Pareto front, we clustered the architectural configurations by using a \(k\)-means clustering algorithm applied to the frame-rate metric to create two clusters of architectural configurations, simply representing low and high frame-rate solutions.

In this case, Figure 7 shows the scatter plot of the clustered configurations with cluster centroids centered around the values of the \(Q\text{-frame rate}\) equal to 1014.1 and 85 mm\(^2\).
A Variability-Aware Design Space Exploration Methodology for On-Chip Multiprocessors

Fig. 7. Final Pareto set clustered in two sets.

Table III. Final Customized Architectures for the MPEG Decoder

<table>
<thead>
<tr>
<th>Parameter/metric</th>
<th>Cluster 0</th>
<th>Cluster 1</th>
</tr>
</thead>
<tbody>
<tr>
<td># Processors</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Processor issue width</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L1 instruction cache size</td>
<td>2K</td>
<td>16K</td>
</tr>
<tr>
<td>L1 data cache size</td>
<td>8K</td>
<td>8K</td>
</tr>
<tr>
<td>L2 private cache size</td>
<td>64K</td>
<td>128K</td>
</tr>
<tr>
<td>L1 instruction cache assoc.</td>
<td>2w</td>
<td>1w</td>
</tr>
<tr>
<td>L1 data cache assoc.</td>
<td>8w</td>
<td>8w</td>
</tr>
<tr>
<td>L2 private cache assoc.</td>
<td>4w</td>
<td>2w</td>
</tr>
<tr>
<td>I/D/L2 block size</td>
<td>16B</td>
<td>16B</td>
</tr>
<tr>
<td>total system area [mm²]</td>
<td>37.5</td>
<td>40.9</td>
</tr>
<tr>
<td>Q_energy_per_frame [J/f]−2</td>
<td>2.27</td>
<td>2.38</td>
</tr>
<tr>
<td>Q_frame_rate [fps]²</td>
<td>1096.2</td>
<td>1250.3</td>
</tr>
<tr>
<td>Parametric Yield</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>frame rate (μ, σ) [fps]</td>
<td>32.5, 2.7</td>
<td>35.7, 2.7</td>
</tr>
<tr>
<td>power density (μ, σ) [W/cm²]</td>
<td>56.74, 1.9</td>
<td>54.95, 1.9</td>
</tr>
<tr>
<td>power (μ, σ) [W]</td>
<td>21.3, 0.7</td>
<td>22.5, 0.8</td>
</tr>
</tbody>
</table>

1279.2, roughly corresponding to an average frame rate of 31.8 fps and 35.7 fps respectively. Then, we selected a ‘golden’ solution for each cluster by using a decision-making-mechanism based on the minimization of the following expression:

\[
\frac{\text{total system area}(\hat{x})}{Q_{\text{energy per frame}}(\hat{x}) \times Y(\hat{x})}.
\]

The two selected system configurations found are shown in Table III. First of all, we can note that the configurations found are very similar to the clustering centroids in...
In terms of frame-rate, representing low and high-performance architectures. We also note that the frame-rate is a monotonic function of the area, while the energy per frame is at a relatively constant value.

The multiprocessor configurations differ mainly in terms of instruction cache and level 2 cache size and associativity, impacting slightly the yield of the constraints ($\Delta = 4\%$). The configuration with lower yield has, however, a significant lower area occupation ($-9\%$) while the frame rate is significantly impacted by process variations in both configurations (standard deviation up to $2.7$ fps). However, the energy per frame ratio is significantly higher for the bigger configuration ($8\%$). In our opinion, the similarities we found between the two selected configuration both in terms of architectures and metrics are mainly due to the tight constraints fixed in the design problems.

More in detail, Figure 8(a) shows the distributions of power density and frame rate. The data have been generated by Monte Carlo simulations based on the model presented in this article. The smaller configuration has, overall, a higher power density distribution than the bigger one, while the standard deviations are similar. The lower yield for the smaller configuration is a direct consequence of this behavior.

Our Monte Carlo sampling phase allows us to compute the quality function and yield associated with each configuration. For each configuration, the random distributions have been sampled 100 times ($N = 100$) to derive a coarse-grained approximation. However, since the sampling is performed on purely analytical distributions, it did not introduce any significant overhead to the overall exploration process.

Figure 8(b) shows the distribution of the overall power consumption and the frame rate. The distributions of the two configurations are partially overlapped. We use an ellipse centered at the average of the two distributions to indicate the $1-\sigma$ boundary. In this case, while the smaller configuration has a bigger power density, the overall power consumption is noticeably lower than the bigger configuration. However, this is not an indication of a more efficient usage of the processor resources. As we mentioned before, the bigger configuration has a better energy per frame ratio thus it decodes faster and with less total energy than the smaller one. The standard deviation of the biggest configuration is however greater than the smaller one, because, as the area increases, the effect of manufacturing process variations on power consumption is more evident.

Moreover, Figure 8 helps in understanding why the selected configurations for each cluster (both c-0 and c-1, see Table III) present an average value of frame rate much bigger than the required 25fps. In fact, given the distribution probabilities associated with process variations, a nominal value of 25fps is not enough to guarantee that
the lower distribution tails meet the constraints. Indeed, a nominal value of 25fps would have given a relatively poor yield (50%) since only the chips associated with the upper distribution tail would have met the performance constraints. Both c-0 and c-1 present an average value of frame rate much bigger than the required 25fps because our algorithm maximized the comprehensive yield (i.e., also of the lower distribution tail) instead of the plain nominal frame rate.

Comparison with conventional approaches. In this section, we compare our modeling and exploration methodology with respect to a conventional approach where the optimization problem has been formulated by imposing sharp constraints on the application requirements and by substituting the quality measures with a direct evaluation of the frame-rate and energy-per-frame metrics. The constraints have been tightened by considering the expected standard deviation on the frame rate, power density and average power consumption.

The conventional search has been run by considering all the feasible configurations of the design space by using a set of tightened constraints and plain system-level metrics instead of quality functions.

We started by considering 3-σ variation on the average values of the constraints but the exploration tool did not find any feasible solution over the entire design space. Reducing the considered variation on the average values of the constraints to 2-σ, the explorer found only 4 configurations (instead of 11 configurations found with our methodology). One out of the 4 configurations corresponds to the configuration c-1 previously found, while the other 3 configurations have similar area occupation (∼40mm²) and yield (∼0.99) but show a very suboptimal behavior in terms of frame-rate and energy-per-frame.

In this article, we considered only 3-σ and 2-σ variations in order to maximize the yield resulting from the conventional approach. While minimizing the σ variation may result in nominally better solutions, it would likely decrease the overall yield. A well-known rule in production quality control is that, if the considered distributions are normal (or log-normal as in our case), a 1-sigma variation would increase more than 1 order of magnitude the number of chips not meeting the application constraints.

Figure 9 shows the area occupation and the quality measures associated with the conventional configurations (named k-0,1,2,3). The quality values have been estimated with the proposed variability-aware model. As can be seen, the quality measures of k-0,1,2 are significantly lower than configuration c-1/k-3 (from 12% up to 20%), showing a clear suboptimal behavior in terms of both frame-rate and energy-per-frame.
Overall, the conventional approach found only a reduced number of feasible configurations compared to our methodology (4 instead of 11). All but one of the feasible conventional configurations are dominated by the configurations found with our approach.

6.2 Embedded PC Processor

This use case considers a system with a wider set of target applications (which can be roughly defined as general purpose) compared to the previous use-case. However, it is characterized by very tight constraints in terms of power density, power consumption and overall die cost such as those required in an industrial embedded deployment scenario.

The target applications used to model the target workload are representative of a general-purpose scenario. They consist of a subset of the SPLASH-2 parallel benchmark suite (fft, lu, cholesky, radix, barnes, fmm, ocean and volrend).

For each application, we gather information about the execution cycles and energy consumption by combining them to obtain the MIPS performance and mW-per-MIPS. Then we combine those figures with a geometric average over the set $U$ of benchmark codes:

$$\text{avg}_{\text{MIPS}}(\vec{x}) = \prod_{\xi_k \in U} \text{MIPS}(\vec{x}, \xi_k)^{\frac{1}{|U|}}$$

(20)

and

$$\text{avg}_{\text{mW-per-MIPS}}(\vec{x}) = \prod_{\xi_k \in U} \text{mW-per-MIPS}(\vec{x}, \xi_k)^{\frac{1}{|U|}}.$$ 

(21)

Averaged MIPS and mW-per-MIPS are then fed into the Monte Carlo estimation framework for computing process variability dependent information in terms of quality values, as described before. We formulated the multiobjective problem as a "robust" problem consisting of finding a system configuration which maximizes the quality values and the overall yield on the system constraints:

$$\max_{\vec{x} \in \mathcal{X}} \begin{bmatrix} \frac{1}{\text{total system area}(\vec{x})} Q_{\text{avg}_{\text{MIPS}}(\vec{x})} \\ Q_{\text{avg}_{\text{mW-per-MIPS}}(\vec{x})} \\ Y(\vec{x}) \end{bmatrix},$$

(22)

where the parametric yield $Y$ is defined as:

$$Y = \text{prob} \left[ \text{avg}_{\text{MIPS}}(\vec{x}) \geq 3.000, P_{\text{dens}} \leq 75 \text{W/cm}^2, P \leq 120 \text{W} \right]$$

(23)

subject to the following constraints:

$$Y(\vec{x}) \geq 0.85, \quad \text{total system area}(\vec{x}) \leq 200 \text{mm}^2$$

(24)

The application requirements in Equation (23) are given by the following.

— The minimum performance requirement. This is introduced as a Quality of Service (QoS) constraint considering a minimum of 3,000 MIPS.

— The average power density. Following ITRS [Kahng 2002], we consider an upper bound of 75W/cm$^2$ for the maximum power dissipation of cost-effective packaging and forced-air cooling.

— The average power dissipation. This constraint is again given by a cost-effective cooling envelope.
As in the previous case study, the area constraint has been defined to impose an overall upper bound to the cost of manufacturing and packaging.

Application of the variability-aware robust DSE methodology. The randomized design of experiments generated as starting point for the optimization methodology a set of 250 simulations, while an additional number of 130 simulations has been generated by the optimization algorithm. Globally, only 6.8% of the target design space has been simulated.

The final Pareto front found by applying the proposed methodology is composed of 59 configurations.

To help the system architect to select among the feasible solutions in the Pareto front, we clustered the architectural configurations by using a $k$-means clustering algorithm applied to the area to generate two clusters (Cluster 0 and 1) of architectural configurations, representing small and big area solutions.

Figure 10 shows the scatter plot of the clustered configurations with the cluster centroids centered around the value of the area equal to 88 and 175 mm$^2$. Then, we selected a “golden” solution for each cluster by using a decision-making-mechanism based on the minimization of the following expression:

$$\frac{Q_{\text{avg. MIPS}}(\vec{x}) \times Q_{\text{avg. mW/MIPS}}(\vec{x}) \times Y(\vec{x})}{\text{total_system_area}(\vec{x})}$$

(25)

The corresponding two system configurations are shown in Table IV. We note that the average MIPS and the mW-per-MIPS is a monotonic function of the area. The CMP configurations mainly differ in terms of number of processors, slightly impacting the yield of the constraints on the MIPS performance ($\Delta = 8\%$). The configuration with lower yield has, however, a significant lower area occupation ($-65\%$). The MIPS, power density and overall power have been significantly impacted by process variations in

A Variability-Aware Design Space Exploration Methodology for On-Chip Multiprocessors
Table IV. Final Champion Architectures for the 2 Clusters

<table>
<thead>
<tr>
<th>Parameter/metric</th>
<th>Cluster 0</th>
<th>Cluster 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Processors</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Processor Issue Width</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>L1 Instruction Cache Size</td>
<td>2K</td>
<td>8K</td>
</tr>
<tr>
<td>L1 Data Cache Size</td>
<td>8K</td>
<td>8K</td>
</tr>
<tr>
<td>L2 Private Cache Size</td>
<td>256K</td>
<td>128K</td>
</tr>
<tr>
<td>L1 Instruction Cache Associativity</td>
<td>1w</td>
<td>1w</td>
</tr>
<tr>
<td>L1 Data Cache Associativity</td>
<td>8w</td>
<td>8w</td>
</tr>
<tr>
<td>L2 Private Cache Associativity</td>
<td>4w</td>
<td>8w</td>
</tr>
<tr>
<td>I/D/L2 Block Size</td>
<td>32B</td>
<td>32B</td>
</tr>
<tr>
<td>Total system area [mm²]</td>
<td>58.75</td>
<td>165.1</td>
</tr>
<tr>
<td>Parametric Yield</td>
<td>0.99</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{avg MIPS} (\mu, \sigma) )</td>
<td>3324, 221</td>
<td>4830, 342</td>
</tr>
<tr>
<td>( \text{avg mW-per-MIPS} (\mu, \sigma) [mW/MIPS] )</td>
<td>8.72, 0.23</td>
<td>15.81, 0.38</td>
</tr>
<tr>
<td>( \text{avg Power} (\mu, \sigma) [W] )</td>
<td>35.32, 3.81</td>
<td>90.16, 9.83</td>
</tr>
<tr>
<td>( \text{avg Power Density} (\mu, \sigma) [W/cm²] )</td>
<td>57.58, 5.87</td>
<td>53.63, 5.57</td>
</tr>
</tbody>
</table>

Fig. 11. MIPS/mW-per-MIPS space (a) and power consumption/MIPS space (b) for the selected configurations of cluster 0 (c-0) and cluster 1 (c-1).

In summary, the results in Table IV derive from a nontrivial decision process which takes into account the parametric yield of the target system. While energy and area increase seems not to be worth the performance increase, the parametric yield grows by a nonnegligible percentage (8%). Moreover, the fact that the performance increase does not follow proportionally both area and energy may also be due to the fact that the target application set does not scale ideally over the number of processors (we note that cluster 1 has 4 times the processors of cluster 0).

More in detail, Figure 11(a) and Figure 11(b) respectively show the variability of MIPS, mW-per-MIPS and overall power. The data have been generated by Monte Carlo simulations based on the model presented in this article. The smaller configuration has, overall, a lower mW-per-MIPS mean than the bigger one while the standard deviations are similar. The configuration with bigger area shows a higher MIPS performance directly impacting the power consumption. The lower yield for the smaller configuration is a direct consequence of the smaller MIPS average value. However, we must point out that this result is to be taken with care, given the high-level models used in this article. The overall yield of a system depends also on the wafer yield which is likely to drop when the chip area increases. The complete assessment of this case...
Comparison with conventional approaches. In order to compare our modeling and exploration methodology with a conventional approach, we formulate the conventional problem by using a set of tighter constraints on the application requirements and by substituting the quality measures with a direct evaluation of the average MIPS, average mW-per-MIPS and average power metrics.

The constraints have been tightened by taking into account the expected standard deviation on all the metrics affected by the process fluctuations. We considered a 3-σ variation on the average values of the constraints. The conventional approach found only two feasible Pareto configurations with respect to the 59 configurations found with the proposed approach.

In Figure 12, we compare the conventional solutions (k-0 and k-1) with the solutions found in the previous sections (c-0 and c-1). In order to give a fair comparison of the two approaches, we grouped the solutions by considering their area occupation and we computed the quality measures of k-0 and k-1 by applying the proposed variability-aware model. The smaller configurations c-0 and k-0 are shown in Figure 12(a), while the bigger configurations are shown in Figure 12(b).

Considering Figure 12(b), we note that configuration c-0 is better than k-0 in terms of area occupation and quality of the mW-per-MIPS figure of merit. The configuration k-0 shows a bigger overall MIPS performance which is directly correlated with the higher area occupation. Considering a geometric average of the three figures of merit, the configuration c-0 results 50% better than k-0.

Considering Figure 12(b), we note that configuration c-1 is slightly bigger than k-1 in terms of area occupation. However, both MIPS and mW-per-MIPS of c-1 are better than k-1 (7 and 3 % respectively). Considering a geometric average of the three figures of merit, the configuration c-1 results 10% better than k-1.

Overall, the conventional approach found only a subset of the feasible solutions (2 out of 59) and 1 out of 2 are suboptimal. This demonstrates that a variability-aware design space exploration is more flexible and efficient than conventional (deterministic) approaches.

7. CONCLUSIONS

In this article, we addressed the problem of variability-aware design at system-level for CMPs, focusing on the impact of manufacturing process variations on the
system-level metrics and consequently on the application-level constraints. This analysis has been demonstrated to be important for systems where meeting throughput or timing constraints is necessary to guarantee the application quality-of-service, such as multimedia and embedded systems.

In particular, we proposed a design space exploration framework which is robust with respect to manufacturing process variations. The main goal is the optimization of the variance of the target system metrics and the maximization of the yield of the system with respect to the application-level constraints. Moreover, we introduced a set of response surface modeling techniques to enable an efficient evaluation of the statistical measures of execution time and energy consumption for each system configuration. The solutions generated by the proposed methodology can further be improved with low level optimization or dynamic compensation, that are orthogonal approaches with respect to the one we proposed, with a reduced effort.

Experimental results on two different case studied provided a comparison of our design space modeling and exploration technique with conventional approaches by highlighting the flexibility and efficiency issues.

REFERENCES


A Variability-Aware Design Space Exploration Methodology for On-Chip Multiprocessors


Received February 2009, revised October 2009, accepted March 2010