

Variability-Aware Robust Design Space Exploration of Chip Multiprocessor Architectures

Gianluca Palermo, Cristina Silvano, Vittorio Zaccaria

Politecnico di Milano - Dipartimento di Elettronica e Informazione

E-mail: {gpalermo, silvano, zaccaria}@elet.polimi.it

Abstract—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 introduce response surface modeling techniques to enable an efficient evaluation of the statistical measures of execution time and energy consumption for each system configuration. We then introduce 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.¹

I. INTRODUCTION

Process variation is dramatically becoming one of the most important challenges related to power and performance optimization for sub-90 nm CMOS technologies. Parametric yield, i.e., 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 inter-die and intra-die variations. Inter and intra-die 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. Inter-die 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, intra-die parameter fluctuations consist of both random and systematic components and generate non-uniform electrical characteristics across the chip [1].

In this scenario, we address the problem of variability-aware design at system-level for chip multi-processors (CMP). More precisely, we tackle the problem of the impact of manufacturing process variations onto the system-level metrics and consequently onto the application-level constraints.

The main contribution of this paper is twofold:

- We introduce a design space exploration (DSE) framework which is robust with respect to manufacturing process variations. The main goal is the optimization of the dispersion of the target system metrics and the maximization of the yield of the system with respect to the application-level constraints.
- The exploration process is supported by response surface modeling (RSM) techniques for improving the overall estimation time to obtain the system-level metrics associated to each system configuration.

The DSE framework is based on a set of state-of-the-art accurate performance, area and energy models of a CMP

¹This work was supported in part by the EC under grant MULTICUBE FP7-216693

TABLE I
PROJECTED VALUES FOR NOMINAL CRITICAL PATHS.

Technology generation (nm)	180	130	100	70	50
Gate length (nm)	140	85	65	45	32
Nominal CP delay (D_{nom}) (ns)	0.8	0.78	0.6	0.42	0.3
CP number (n)	10^2	10^3	10^3	10^4	10^4

taking into account process variations at the 70nm technology node. The CMP architecture to be explored is composed of a variable number of out-of-order processors with private L1 and L2 caches. To estimate system-level metrics, we leveraged the SESC [2] simulation tool, a fast MIPS instruction set simulator for CMPs providing dynamic energy and execution cycles measures associated to the execution of the target application. Area estimation has been carried out by using the models proposed in [3]. Although in this paper the SESC simulator has been used, the proposed framework is more general and easily retargetable to other multi-processor architectures and technology nodes.

The paper is organized as follows. Section II briefly introduces the background on reference, state-of-the-art performance and power models used for estimating the impact of manufacturing process variations on the systems figure of merit. Section III introduces the response surface models for energy and delay while Section IV introduces the design space exploration methodology proposed in this paper. Section V shows the experimental results of the proposed methodology.

II. BACKGROUND ON VARIABILITY AWARE DELAY AND ENERGY MODELS FOR SYSTEMS-ON-CHIP

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 number of cycles of execution can in turn depend on the critical path delay in the worst case (as in the case of asynchronous multi-processors). The maximum critical path delay can be assumed (as suggested in [1], [4]) 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 I shows the reference nominal values for the critical path delay [5] 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 [1].

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 intra-die and inter-die components [1]. The total probability distribution of

the critical path delay D is the following:

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

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

Previous works [6], [1] proposed some approximations for Equation 1. Based on [1], 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 σ_h^2 . According to [1], [7], [4] 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:

$$P = P_{switching} + P_{leak} \quad (2)$$

The term $P_{switching}$ is the power consumption of the system due to switching activity of the internal nodes and the capacity driven by those nodes. Based on [4], [8], [9], we can assume that $P_{switching}$ varies with the maximum allowable frequency and it is dependent on the total switching capacitance $\sum_i C_i$ for the target technology node.

The leakage power P_{leak} is the power due to static leakage currents [9]. Following [4], [9] we can assume that the leakage power has a log-normal distribution:

$$\mu_{log,leak} = \ln(\mu_{leak}) - \frac{1}{2} \ln \left(1 + \frac{\sigma_{leak}^2}{\mu_{leak}^2} \right) \quad (3)$$

$$\sigma_{log,leak}^2 = \ln \left(1 + \frac{\sigma_{leak}^2}{\mu_{leak}^2} \right) \quad (4)$$

According to [4], [10], [11], we can use a projected value of leakage power density around ($3W/cm^2$) 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 of information on the specific behavior. For this reason we consider a conservative upper-bound σ_{leak}/μ_{leak} ratio of 20% projected from data in [4].

The measure of the yield has been defined as the percentage of the dies which do not violate the set of constraints imposed on the architecture. In a typical case, the yield can be defined in terms of power P and critical path delay D as follows [4]:

$$Y = Pr[D \leq D_0, P \leq P_0] \quad (5)$$

Both D and $\ln(P_{leak})$ terms follow a bi-variate Gaussian distribution and they are inversely correlated by a correlation factor ρ , thus the yield cannot be computed as a simple product of probabilities. Although the actual layout of the processor is really important for determining the coefficient of correlation, we actually lack of specific configuration-dependent information for ρ . Nevertheless, in the current literature, typical values of ρ range from -0.65 to -0.95 (average -0.87), as reported in [4]. We will assume an average value of $\rho = -0.87$, while

the problem of considering a configuration dependent ρ could be the subject for future research.

The state-of-the-art energy and delay models reported in this section have been integrated into our variability-aware DSE framework for CMPs.

III. EFFICIENT EVALUATION OF THE SYSTEM METRICS USING RESPONSE SURFACE MODELING

The proposed reference models pose serious challenges to the overall DSE framework because of, for each architectural configuration, the designer should sample the correlated power and delay distributions to obtain a comprehensive view of the architecture behavior. Each sample, however, should also be evaluated in terms of dynamic contribution to the energy and execution time delay, which depend on the particular application executed on the architecture. This approach would force the designer to execute a huge number of simulations.

To tackle this complexity, we introduce a set of response surface models based on linear regression to capture the dynamic behavior of the application on the target architecture by performing as few simulations as possible.

Although it is likely that, in an asynchronous scenario, at many points of concurrent execution of the application some core is being stalled due to sort of thread synchronization, we limit our analysis on synchronous multi-processors. In such a scenario, it is reasonable to estimate the latency of the application in terms of number of required cycles which do not depend on process variation, because making processors faster or slower (for e.g. due to variation) does not create totally different synchronization points.

Linear regression is a well-known regression method that models a linear relationship between a dependent response function f and some independent variables x_i ($i = 1 \dots p$) plus a random term ε . Such type of model contains both simple as well as exponential terms combined linearly. However, research efforts [12] have mainly been addressed to first and second order models. The general expression for a second order linear surface model is given by:

$$f(\vec{x}) = \alpha_0 + \sum_{k=1}^p \alpha_k x_k^2 + \sum_{i=1}^p \sum_{k=1, k \neq i}^p \beta_{i,k} x_k x_i + \sum_{k=1}^p \gamma_k x_k + \varepsilon \quad (6)$$

Least squares analysis can be used to determine a suitable estimate for the coefficients α, β, γ . Least squares analysis determines the values of unknown quantities in a statistical model by minimizing the sum of the squared residuals (i.e. the differences between predicted and observed values). A measure of the *quality of fit* associated with the resulting model is called *coefficient of determination* $R^2 = \frac{SSR}{SST}$ where SST is the total sum of squares of observations y_i and average observations \bar{y} , while SSR is the regression sum of squares between the model estimates f_i and the average of observations. 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.

When tuning a linear response surface model, the first step is to assess the correct model order (first or second) and the type of parameters x_i (or *predictors*) to be considered. To select the most relevant parameters, the *main effect* analysis [13] on a random subset of the design space configurations can be applied. Assuming a decomposition of the set of parameters into *relevant*, *slightly-relevant* and *irrelevant* parameters, we

TABLE II

COEFFICIENT OF DETERMINATION FOR THE FIVE BEST-IN-CLASS MODELS

Model order	2 nd	2 nd	1 st	1 st	2 nd
Model configuration	heavy	medium	heavy	medium	light
R^2	0.944	0.88	0.73	0.72	0.54

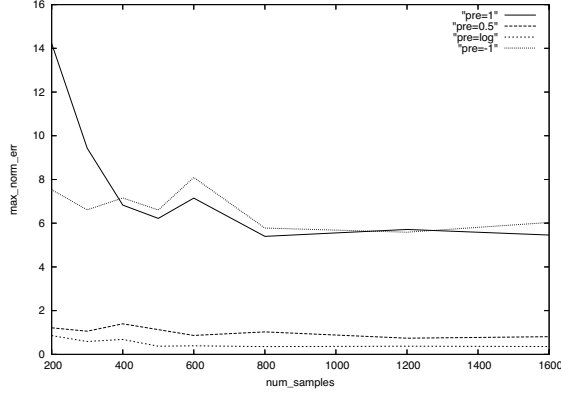


Fig. 1. Maximum normalized error of the linear regression model on the training set by varying the number of input configurations for the selected Box-Cox preprocessing transformations.

clustered our analysis around three model configurations: *Heavy*: all the design space parameters are included in the model, *Medium*: all the parameters except for the irrelevant are used, *Light*: only the relevant parameters are used.

The following results report the comparison between the estimates of response surface models and simulation-based measurements obtained with SESC in terms of energy and delay metrics at system-level. For this comparison, we selected a representative set of four applications (FFT, OCEAN, LU, RADIX) derived from the SPLASH-2 [14] parallel benchmark suite from Stanford. For each benchmark, three different data-sets have been considered, each one representative of a small/medium/large workload size. Table II shows the five best models in terms of coefficient of determination. For our experimental results, the second-order heavy model has been chosen.

The next step for the model tuning is the analysis of the effect of Box-Cox [15] preprocessing transformations on the accuracy of linear models. The preprocessing function transforms the response values before being fed to the linear model training in order to minimize the error. We considered a typical set of transformations as potential candidates $\{y^1, y^{0.5}, \log(y)$ and $y^{-1}\}$. For each transformation, we selected a set of random configurations as input to the linear regression model and we derived the maximum normalized error on the training set. As Figure 1 shows, the best behaving output transformation is the $\log(y)$ function that has been chosen as preprocessing function for our experimental results.

Although the proposed response surface modeling approach is similar to [16], we introduced the main effect analysis and Box-Cox transformations, dramatically improving the quality of the model.

IV. A ROBUST DESIGN SPACE EXPLORATION METHODOLOGY

We define as Robust Design Space Exploration (RDSE) framework, a set of optimization techniques with the following goals: I) minimization of the dispersion of the target system metrics due to manufacturing process variability, II) maximization of the yield with respect to application-level constraints. The goals are directly correlated to the fact that overestimating fluctuations impacts on the design effort, and underestimating fluctuations impacts on the manufacturing effort. Moreover, the yield of the target device when deployed in the real world should be maximized in order to minimize the losses.

A naive way of indirectly maximizing the yield during the exploration is to strengthen up the constraints. This approach has however some drawbacks:

- It is difficult to predict which kind of constraint overhead should be introduced in order to optimize the yield; especially because the figures of merit are tightly correlated and could be a complex function of the original source of variations. This process is error prone and could either discard feasible solutions or accept unfeasible solutions.
- It is impossible to trade-off the yield with the improvements on the figures of merit. We will show that our modeling methodology is crucial in providing such flexible optimization methodology.

As suggested in [17], [18], aggregate functions can be used to optimize both mean and variance of a target function which is dependent on a distribution probability. Aggregate functions can take into account several statistical moments at the same time, thus they are suitable for reducing the size of the multi-objective problem to be solved. In our RDSE methodology, we want to minimize both mean and variance of energy and execution time, which are dependent on the manufacturing process variations. To this purpose, some key definitions from the theory of quality design [18], [19] must be introduced.

For the target response y , we define an aggregate quality performance measure $Q_{y,S}$ which is a monotonic function of the *signal-to-noise* ratio S_{N_Y} for *as-small-as-possible* problems [20]. For a specific response y and a set of N samples y_i picked up from the random variable y , the term $Q_{y,S}$ is defined as follows:

$$Q_{y,S} = 10^{\frac{S_{N_Y}}{10}} = \frac{1}{\left(\frac{1}{N} \sum_{i=1}^N y_i^2\right)} \quad (7)$$

It can be proved that the defined performance metric is an aggregate quality performance measurement of mean and variance [21]. As for *as-small-as-possible* problems, we can define $Q_{y,L}$ as a quality measure for *as-large-as-possible* problems:

$$Q_{y,L} = 10^{\frac{S_{N_Y}}{10}} = \frac{1}{\left(\frac{1}{N} \sum_{i=1}^N \frac{1}{y_i^2}\right)} \quad (8)$$

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.

A. Multi-Objective Exploration Algorithm

The proposed multi-objective optimization meta-heuristic leverages a random design of experiments technique coupled with a linear regression RSM. Recent studies [22] confirmed that, in the field of multi-processor DSE, a randomized design of experiments enables the best tuning of response surface models.

The proposed design flow is basically composed of a loop controlled by the exploration kernel which generates candidate configurations for building the Pareto front of the system configurations. The Pareto front is the set of configurations which are optimal from the point of view of the quality and yield optimization problem:

$$\max_{\vec{x} \in X} \begin{bmatrix} Q_1(\vec{x}) \\ \vdots \\ Q_m(\vec{x}) \\ Y(\vec{x}) \end{bmatrix} \quad (9)$$

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

The multi-objective exploration algorithm 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: $\hat{P}_1 = \text{Pareto}(\hat{S}_1)$.
- 5) Run the simulations to derive the *actual measurements* on the architectural configurations contained in \hat{P}_1 . The result is the design space P_1 .
- 6) If P_1 covers P_0 ($C(P_1, P_0)$) 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 using a decision-making-mechanism. This approach has been shown in the experimental results section.

TABLE III

DESIGN SPACE FOR THE SHARED-MEMORY MULTI-PROCESSOR PLATFORM

Parameter	Min.	Max.
# Processors	2	16
Processor issue width.	1	8
L1 instruction cache size	2K	16K
L1 data cache size	2K	16K
L2 private cache size	32K	256K
L1 instruction cache assoc.	1w	8w
L1 data cache assoc.	1w	8w
L2 private cache assoc.	1w	8w
I/D/L2 block size	16	32

V. EXPERIMENTAL RESULTS, AN 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. The target architecture is a shared-memory multiprocessor with private L2 cache. We focused our analysis on the architectural parameters listed in Table III, where the minimum and maximum values have been reported. Globally, the resulting design space consists of 2^{17} alternative configurations.

For this application-specific customization, the multi-objective optimization problem has been formalized as follows:

$$\max_{\vec{x} \in X} \begin{bmatrix} \frac{1}{\text{total_system_area}(\vec{x})} \\ Q_{\text{energy_per_frame}}(\vec{x}) \\ Q_{\text{frame_rate}}(\vec{x}) \\ Y(\vec{x}) \end{bmatrix} \quad (10)$$

where the yield has been defined as:

$$Y = \text{Pr}[\text{frame_rate}(\vec{x}) \geq 25, P_{\text{dens}} \leq 60\text{W}/\text{cm}^2, P \leq 25\text{W}] \quad (11)$$

and subject to the following constraints:

$$\text{total_system_area}(\vec{x}) \leq 85\text{mm}^2 \quad (12)$$

$$Y(\vec{x}) \geq 0.85 \quad (13)$$

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:

- 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 [23] we consider an upper bound of $60\text{W}/\text{cm}^2$ for the maximum power dissipation of cost-effective packaging and forced-air cooling.
- The average power dissipation. This constraint is given by desired battery lifetime.

The area constraint (Equation 12) has been defined to impose an overall upper bound to the cost of manufacturing and packaging.

In our case study, we used the ALPBench MPEG2decoder [24]. The objective functions have been derived as a geometric average of the system metrics over a set of five input datasets composed of 10 frames at a resolution of 640x480. The randomized design of experiments generated as starting point for the optimization methodology a set of 250 simulations,

while an additional number of 70 simulations has been generated by the optimization algorithm. Globally, only 2.5% of the target design space has been simulated. The final approximated Pareto front is composed of 11 configurations.

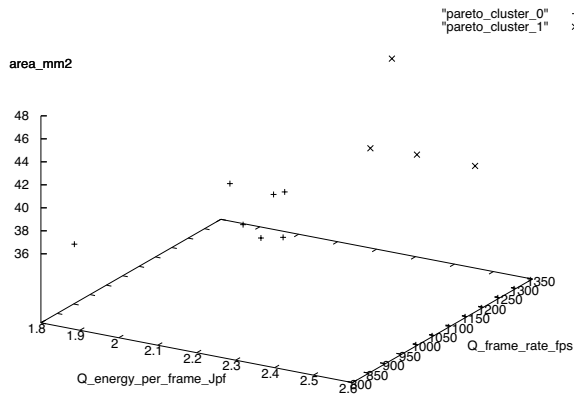


Fig. 2. Final Pareto set clustered in two sets.

To help the system architect to select among the feasible solutions in the Pareto front, we can envision an approach that starts by clustering the architectural configurations and then selects a 'golden' solution for each cluster by using a decision-making-mechanism. In this case, we used 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. Figure 2 shows the scatter plot of the clustered configurations with cluster centroids centered around the values of the Q -frame-rate equal to 1014.1 and 1279.2, roughly corresponding to an average frame rate of 31.8 fps and 35.71 fps respectively. For each cluster, we identified the best configuration by minimizing the expression:

$$\frac{\text{total_system_area}(\vec{x})}{Q_energy_per_frame(\vec{x}) * Y(\vec{x})} \quad (14)$$

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

The multi-processor 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 quality ratio is significantly higher for the bigger configuration (8%).

More in detail, Figure 3 shows the yield of power density and frame rate. The data have been generated by Monte-Carlo simulations based on the model presented in this paper. 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.

TABLE IV

FINAL CUSTOMIZED ARCHITECTURES FOR THE MPEG DECODER.

Parameter/metric	Cluster 0	Cluster 1
# Processors	4	4
Processor issue width.	1	1
L1 instruction cache size	2K	16K
L1 data cache size	8K	8K
L2 private cache size	64K	128K
L1 instruction cache assoc.	2w	1w
L1 data cache assoc.	8w	8w
L2 private cache assoc.	4w	2w
I/D/L2 block size	16B	16B
total_system_area [mm^2]	37.5	40.9
Q_energy_per_frame [J/f] ⁽⁻²⁾	2.3	2.5
Q_frame_rate [fps] ²	1036.2	1253.2
Yield	0.95	0.99
frame rate (μ, σ) [fps]	32.5, 2.7	35.7, 2.7
power density (μ, σ) [W/cm^2]	56.74, 1.9	54.95, 1.9
power (μ, σ) [W]	21.3, 0.7	22.5, 0.8

Figure 4 shows the yield of the overall power consumption and the frame rate. The distributions of the two configurations are partially overlapped. We use a circle centered at the average of the two distributions to indicate the $1\text{-}\sigma$ 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 quality 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 of, as the area increases, the effect of manufacturing process variations on power consumption is more evident.

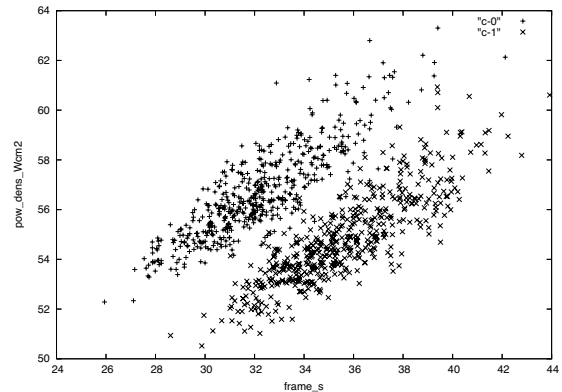


Fig. 3. Yield on power density and frame rate for the selected configurations of cluster 0 (c-0) and cluster 1 (c-1)

A. 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 strengthened by considering the expected standard deviation on the frame rate, power density and average power consumption. We considered both $2\text{-}\sigma$ and $3\text{-}\sigma$ variation on the average values of the constraints.

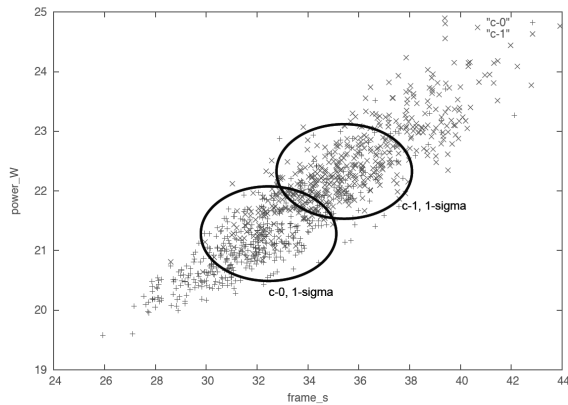


Fig. 4. Yield on power consumption and frame rate for the selected configurations

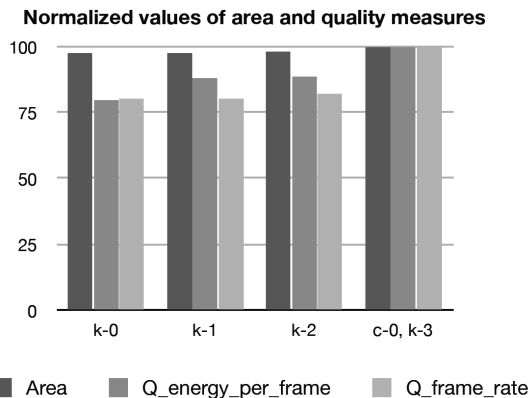


Fig. 5. Area and quality measures for the configurations found by the conventional approach.

Considering the $3\text{-}\sigma$ case, the exploration tool did not find any feasible solution over the entire design space. However, in the $2\text{-}\sigma$ case, 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. The other 3 configurations have similar area occupation ($\sim 40\text{mm}^2$) and yield (~ 0.99) but show a very sub-optimal behavior in terms of frame-rate and energy-per-frame.

Figure 5 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 sub-optimality behavior in terms of both frame-rate and energy-per-frame.

Overall, the conventional approach found only a reduced number of feasible configurations with respect to our methodology (4 instead of 11). All but one of the feasible conventional configurations are dominated by the configurations found with our approach.

VI. CONCLUSIONS

In this paper, 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. Then, we proposed a design space exploration framework which is robust with respect to manufacturing process variations. The main goal is the optimization of the dispersion of the target system metrics and the maximization of the yield of the system with respect to the application-level constraints. Experimental results provided a comparison of our design space modeling and exploration technique with conventional approaches by highlighting its flexibility and efficiency.

REFERENCES

- [1] K Bowman, S Duvall, and J Meindl. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *IEEE Journal of Solid-State Circuits*, pages 183–190, Jan 2002.
- [2] J. Renau et al. SESC simulator, January 2005. <http://secc.sourceforge.net>.
- [3] Ron Kalla, Balam Sinharoy, and Joel M. Tendler. Ibm power5 chip: A dual-core multithreaded processor. *IEEE Micro*, 24(2):40–47, 2004.
- [4] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director. Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance. *Proceedings of the 42nd DAC*, 2005.
- [5] Cheming Hu, Mark Horowitz, and Stephen Y. Chow. Life at the end of cmos scaling (and beyond) (panel session) (abstract only). In Rob A. Rutenbar, editor, *DAC '00: Proceedings of the 37th conference on Design automation*, page 85, New York, NY, USA, 2000. ACM.
- [6] D Marculescu and E Talpes. Variability and energy awareness: a microarchitecture-level perspective. *Proceedings of DAC-42*, pages 11–16, 2005.
- [7] S. Nassif et al. High performance cmos variability in the 65nm regime and beyond. *IEEE International Electron Devices Meeting*, pages 569–571, Nov 2007.
- [8] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. *Proceedings of the 40th Design Automation Conference (DAC 03)*, pages 338–342, Jan 2003.
- [9] R Rao, D Blaauw, D Sylvester, and A Devgan. Modeling and analysis of parametric yield under power and performance constraints. *IEEE Design and Test of Computers*, pages 376–385, Jan 2005.
- [10] E Nowak. Maintaining the benefits of cmos scaling when scaling bogs down. *IBM Journal of Research and Development*, Jan 2002.
- [11] P O'Connor. Future trends in microelectronics - impact on detector readout. *SNIC Symposium, Stanford, California*, pages 1–6, Jan 2006.
- [12] R Myers, A Khuri, and W Carter Jr. Response surface methodology: 1966–1988. *Technometrics*, 31(2):137–157, Jan 1989.
- [13] T. J. Santner, Williams B., and Notz W. *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.
- [14] S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta. Splash-2 programs: characterization and methodological considerations. *Proceedings of the 22th International Symposium on Computer Architecture*, page 2436, 1995.
- [15] P.J Joseph, Kapil Vaswani, and M.J Thazhuthaveetil. Construction and use of linear regression models for processor performance analysis. *High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*, pages 99–108, 2006.
- [16] B. Lee and D. Brooks. Accurate and efficient regression modeling for microarchitectural performance and power prediction. *ACM SIGOPS Operating Systems Review*, 40(5):185–194, Oct 2006.
- [17] H Beyer and B Sendhoff. Robust optimization—a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33–34):3190–3218, Jan 2007. Accepted for Publication.
- [18] G. Taguchi. *The System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*. Quality Resources, 1987.
- [19] A Song, A Mathur, and K Pattipati. Design of process parameters using robust design techniques and multiple criteria optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 25(11), Jan 1995.
- [20] G Box. Signal-to-noise ratios, performance criteria, and transformations. *Technometrics*, 30(1):1–17, Jan 1988.
- [21] K Lee and G Park. Robust optimization in discrete design space for constrained problems. *AIAA Journal*, 40(4):774–780, Jan 2002.
- [22] G. Palermo, C. Silvano, and V. Zaccaria. An efficient design space exploration methodology for multiprocessor soc architectures based on response surface methods. In *Proceedings of SAMOS08*, 2008.
- [23] A Kahng. Directions for drivers and design. *Circuits and Devices Magazine, IEEE*, 18(4):32–39, Jul 2002.
- [24] Man-Lap Li, R Sasanka, S Adve, Yen-Kuang Chen, and E Debes. The alpbench benchmark suite for complex multimedia applications. *Workload Characterization Symposium, 2005. Proceedings of the IEEE International*, pages 34–45, Sep 2005.