Impact of Uncertainty and Correlations on Mapping of Embedded Systems

Wenkai Guan, Milad Ghorbani Moghaddam and Cristinel Ababei
Department of Electrical and Computer Engineering
Marquette University
Email: {wenkai.guan, milad.ghorbanimoghaddam, cristinel.ababei}@marquette.edu

Abstract—**The impact of multiple levels of uncertainty in design parameters and uncertainty correlations on the quality of mapping solutions in embedded systems is investigated. The investigation is done with a simulation tool developed to conduct multi-objective design space exploration in order to generate robust Pareto frontiers in the solution space formed by reliability, execution time, and energy as design objectives. The simulation tool integrates proposed models for uncertainty and a hypervolume based technique for quantifying the difference between Pareto frontiers. Simulations results show that by not considering uncertainty and correlations between different sources of uncertainty can lead to overestimation of the performance of the optimal solutions.**

I. Introduction

New challenges in embedded systems design include uncertainty in design parameters due to process, voltage and temperature (PVT) variations among others [1]. The uncertainty in design parameters adversely affects the accuracy of the model-based estimations of the design attributes of interest, thereby affecting the quality of the final solutions. Thus, increased uncertainties in design parameters undermine the accuracy and effectiveness of the mapping of applications to architecture platforms in embedded systems design. Therefore, researchers started to address the issue of uncertainties in design parameters during the process of designing embedded systems. For example, several recent studies presented solutions to model and handle uncertainty [2]–[4]. The work in [2] proposed uncertainty models and a mapping algorithm for robust embedded systems that used software components only. Reliability was the only objective considered in the optimization process. The study in [3] proposed an uncertainty-aware reliability model for the design space exploration of embedded systems with consideration of correlations between components. Similarly to [2], reliability was considered as the only uncertain design attribute. The work in [4] used the uncertainty and reliability models from [2] and proposed a solution to the mapping problem similar to that from [5], [6], but formulated as a multi-objective problem considering reliability, execution time, and energy consumption together.

This paper extends the work in [4] by proposing models for multiple correlated sources of different levels of uncertainty. In addition to these models, the developed custom simulation tool integrates a hypervolume based technique for quantifying the difference between Pareto frontiers. Thus, the main contribution of this work is the consideration of multiple correlated sources of uncertainty into the formulation of the mapping problem as a multi-objective optimization problem, which is solved with a genetic algorithm that is described next.

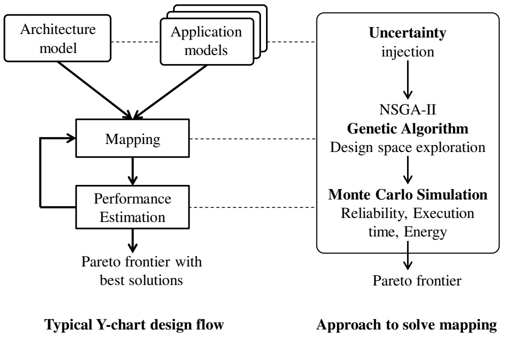

Fig. 1. Mapping approach from [4], which is extended in this paper.

II. Mapping of Embedded Systems with Consideration of Uncertainty

A. Design Space Exploration with NSGA-II Genetic Algorithm

The tool from [4], extended in this paper, addresses the problem of mapping in embedded systems. Finding the best mapping solution is done with an iterative optimization algorithm that conducts an automated design space exploration. The process follows a Y-chart design flow as illustrated in Fig. 1. In this iterative process, the application tasks and communications between them are assigned or mapped to components of the hardware architecture platform. This mapping process is formulated as a multi-objective problem, which considers reliability, execution time, and energy as the three main objectives. To solve this multi-objective mapping problem, the Non-dominated Sorting Genetic Algorithm (NSGA-II) [7] is used due to 1) its ability to handle multiple objectives at the same time, 2) its efficiency because of the lower computational complexity, and 3) the ease of implementation. The NSGA-II step (Fig. 1) represents the main iterative loop where the mapping *solution space* is explored.

B. Monte Carlo Simulation for Design Attributes Estimation

The Monte Carlo (MC) simulation step from Fig. 1 is used to evaluate each solution explored during the search process. During the MC simulation, a given mapping solution candidate is evaluated for different values of the parameters affected by uncertainties. The evaluation consists of the estimation of reliability, execution time, and energy consumption using

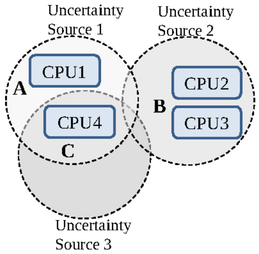

Fig. 2. Illustration to aid in the definition of different correlation groups. Circles denote the sphere of influence of a given uncertainty source.

specific models. Reliability is estimated using absorbing Discrete Time Markov Chain (DTMC) models adopted from [8]. Execution time and energy consumption are estimated using specific equations described in [4].

C. Uncertainty Modeling

When modeling the uncertainty in design parameters in embedded systems, one needs to consider the uncertainty correlations. That is because multiple uncertainty sources (e.g., heat sources) may affect several components simultaneously. Thus, a correlation between uncertainties of components may exist. The work in [3] is a recent attempt to consider such correlations when modeling uncertainties. We model the uncertainty correlations in a similar way, but with the difference that we consider the uncertainty correlations under the influence of multiple uncertainty sources of different levels or degrees of uncertainty. First, we introduce some definitions with the help of the illustration from Fig. 2:

Correlation group: A set of components of the architecture model that are affected by one or more uncertainty sources.

Independent group: If the correlation group *g* contains only one component affected by one uncertainty source, then, we call group *g* an independent group. An example is group A formed by the component CPU1 in Fig. 2.

Single correlation group: If all the components in the correlation group *g* are affected by the same uncertainty source, then, we call group *g* a single correlation group. An example is group B formed by the components CPU2 and CPU3 in Fig. 2.

Multiple correlations group: If different components in the correlation group *g* are affected by multiple uncertainty sources, then, we call group *g* a multiple correlations group. An example is group C formed by the component CPU4 in Fig. 2.

q_g^{th} *percentile:* is the percentile we sample in correlation group *g* from the probability distributions of the uncertain parameters [3]. It is used to guarantee that all components in group *g* vary together for each sampling process.

In order to model uncertainty correlations, the sampling process inside the Monte Carlo simulation step (Fig. 1) is modified as follows:

1) For an independent group, samples from the uncertain design parameters probability distributions are generated using independently sampled parameters.
2) For a single correlation group, samples from the uncertain design parameters are generated using the q_g^{th} percentile of its probability distribution, where q_g is a uniformly distributed random number that satisfies $q_g \in [0, 1]$ for each sampling process.
3) For a multiple correlations group, samples from the probability distributions of the uncertain design parameters are generated using the q_g^{th} percentile of its probability distribution, where q_g^{th} is determined using the following equation:

$$q_g^{th} = \sum_{i=1}^{n} \alpha_i \cdot q_g^i \tag{1}$$

where α_i is a coefficient in [0,1], which is used to measure the degree of influence from the uncertainty source i to group g, and it satisfies $\sum_{i=1}^{n} \alpha_i = 1$. q_g^i is the q_g^{th} percentile for group *g* affected by uncertainty source i.

By generating samples in this way, the uncertain parameters from the components in a single correlation group vary together, and their variations are independent of those from different single correlation groups. Also, the samples of the uncertain parameters for components in multiple correlations groups combine the influence of all uncertainty sources, and such combinations are independent of those of other multiple correlation groups. With the concept of correlation groups and the uncertainty modeling modified as above, one can easily consider also different levels of uncertainty. Different levels of uncertainty are injected into the design parameters of the components according to the assumed degree of the influence of the uncertainty sources forming the correlation groups.

III. Simulation Results

The uncertainty model presented in the previous section was implemented into a custom software tool, which follows the steps in Fig. 1. This tool is referred to as the *New* tool. We conduct comparisons to the traditional Sesame tool [9], [10] as well as to the approaches from [3], [4]. As an additional contribution, the New tool uses OpenMP to speedup the computational runtime of the simulations. All simulations are done on a 64 bit Intel i5-4690 CPU, 3.50 GHz x4 with 8 GB memory running the Ubuntu 14.04 LTS operation system.

Each of the compared mapping approaches provides as result a Pareto Frontier (PF) formed by non-dominated solution points (Fig. 1). For comparison purposes, to quantify the difference between such Pareto frontiers, we use the concept of hypervolume indicator introduced in [11]. More specifically, we propose to use the absolute difference between the hypervolume indicators corresponding to two different Pareto frontiers, PF_1 and PF_2:

$$diff(\overline{PF_1}, \overline{PF_2}) = |H(\overline{PF_1}) - H(\overline{PF_2})| \tag{2}$$

where the hypervolume indicator for a normalized PF (e.g., $\overline{PF_1}$ or $\overline{PF_2}$ in our case) is defined as [12]:

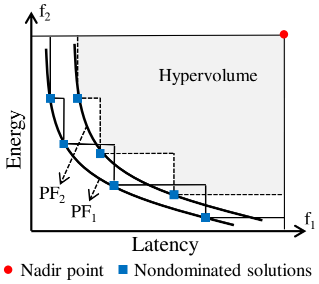

Fig. 3. The difference between hypervolume indicators is used to quantify the difference between the two Pareto frontiers, PF_1 and PF_2.

$$H(\overline{PF}) = \int 1[\exists x \in X : f(x) \prec y \prec z] dy \qquad (3)$$

where f is the vector obtained by stacking the objectives, $\prec$ is the dominance operator defined as $x \prec y \Leftrightarrow (x_1 < y_1) \wedge \ldots \wedge (x_m < y_m)$, and $z \in R^N$ is the reference point, which is called the Nadir point - as the worst-known value in each dimension. For example, Fig. 3 illustrates how the measure in equation (2) quantifies the difference between two normalized Pareto frontiers.

A. Comparison to Traditional Sesame Tool

The comparison with the traditional Sesame tool is done on two testcases, MJPEG and MP3, which are provided with the Sesame tool [9], [10]. We use the same target architecture as the Sesame tool. The architecture includes five different processors and a memory connected to a shared bus. The Sesame tool was also used to collect average execution cycles per task for both testcases; we use these values in the simulations with the New tool for a fair comparison. In addition, we adopt the power consumption values for different processors from [13], which described the parameters of the Sesame architecture platform; thus, the results reported here are as close as possible to 'real system results'.

The normalized Pareto frontiers obtained using Sesame and New tools are shown in Fig. 4 for the MJPEG testcase. A similar plot was obtained for the MP3 testcase; not included here due to lack of space. Fig. 4 shows that the New tool generates Pareto frontiers that are shifted away from those obtained by the Sesame tool. The solution points generated by the Sesame tool have better overall performance (solution points are closer to the center of coordinates of the solution space) than the solutions on the Pareto frontier obtained by the New tool. However, the solutions found by the Sesame tool are unaware of uncertainty and provide an optimistic view of what the performance is. Thus, the traditional deterministic approach of the Sesame tool *overestimates* design attribute values, and this can lead to inaccurate final solutions. The term *overestimate* is used in this context with the meaning that 'performance' is found to be better than what it is, which is reported as smaller energy consumption and shorter latencies.

As discussed earlier, to quantify the difference between the two Pareto frontiers, we calculate the hypervolume indicators and their difference as in equation (2). The result of these calculations is reported in Table I. The difference indicates that the Sesame traditional deterministic approach may lead to significantly inaccurate 'optimal' solutions.

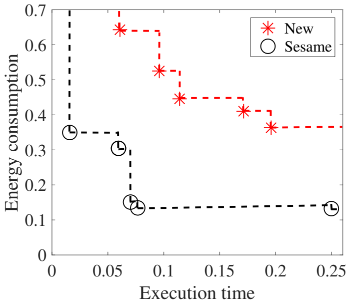

Fig. 4. Pareto frontiers generated by Sesame and New tools for the MJPEG testcase. Axes values are normalized.

TABLE I
HYPERVOLUMES AND THEIR DIFFERENCE FOR THE PARETO FRONTIERS GENERATED BY THE NEW AND SESAME TOOLS.

Testcase	$H(New)$	$H(Sesame)$	**Difference as %** **Comp. to Sesame [9]**
MJPEG	0.5794	0.8421	31.2%
MP3	0.5273	0.9607	45.1%

B. Comparison to Current State-of-the-Art Tools

Here, we compare the New tool with the current state-of-the-art tools, which include the tool in [4] (referred to as *Prev1* tool) and the approach in [3] (referred to as *Prev2* tool), which has been implemented inside the New tool. As a reminder, the Prev1 tool did not consider uncertainty correlations and the Prev2 tool did not consider different levels of uncertainty and multiple correlation groups. In all simulations, we used the architecture platform from [4] that includes five software components (e.g., CPUs), five hardware components (e.g., FPGAs), and two memory components connected to a shared bus. We assume all the software components are affected by the same uncertainty source with injected 10% level of uncertainty, and all the hardware components are influenced by another uncertainty source at 5% level of uncertainty. Therefore, in this way two correlation groups are emulated - according to the discussion earlier in the paper. Simulations were done on four testcases, ABS, ACC, H.264, and JPEG. The results obtained are similar for all testcases; the discussion here focuses on the ABS testcase only in the interest of space.

The Pareto frontiers obtained by the New, Prev1, and Prev2 tools are shown in Fig. 5. We first compare the Pareto frontiers generated by New and Prev1. Fig. 5 shows that the Pareto frontier obtained with the New tool is also shifted away from that generated by the Prev1 tool. The shift amount is proportional to the level of injected uncertainties. This difference between the Pareto frontiers is expected because the Prev1 approach models the uncertainty affecting different components independently, while the New approach considers the uncertainty correlations among the components, which restricts the process of sampling of parameters for all components in the Monte Carlo simulations. Next, we compare

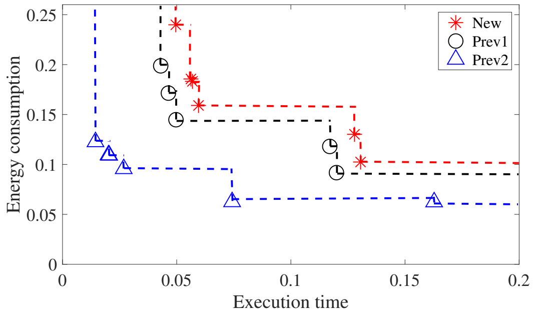

Fig. 5. Pareto frontiers generated by the New, Prev1, and Prev2 tools for the ABS testcase. Axes value are normalized.

the Pareto frontier generated by New with the one generated by Prev2. We again notice a shift difference between the two Pareto frontiers. This difference is also expected because Prev2 did not consider multiple correlation groups and different levels of uncertainty - which ignores the fact that some of the components in multiple correlation groups (e.g., affected by multiple uncertainty sources) may have higher level of uncertainty than other components. We conclude that design objectives in multiple correlation groups may be *overestimated* when using Prev2 for analysis. Note that we cannot directly compare Prev1 with Prev2 because Prev1 did not consider uncertainty correlations while Prev2 did not consider different levels of uncertainty and multiple correlation groups.

Next, to quantify the difference between the Pareto frontiers, we calculate the hypervolume indicators and their difference as in equation (2). The result of these calculations is reported in Table II. Similarly to the discussion in the previous section, we observe that the difference between New and Prev1 is $2.0 - 5.3\%$, while the difference between New and Prev2 is $8.0 - 16.3\%$. These differences again indicate that Prev1 and Prev2 approaches *overestimate* the design objectives and provide better solutions that are inaccurate. In contrast, by considering both uncertainty correlations and different levels of uncertainty, the New tool can provide robust and relatively more accurate solutions than current state-of-the-art tools.

TABLE II
HYPERVOLUMES AND THEIR DIFFERENCE FOR THE PARETO FRONTIERS GENERATED BY THE NEW, PREV1, AND PREV2 TOOLS.

Testcase	$H(New)$	$H(Prev1)$	$H(Prev2)$	**Difference as % Comp. to Prev1 [4]**	**Difference as % Comp. to Prev2 [3]**
ABS	0.8479	0.8649	0.9218	2.0%	8.0%
ACC	0.8243	0.8445	0.9843	2.4%	16.3%
H.264	0.8654	0.8884	0.9626	2.6%	10.1%
JPEG	0.7631	0.806	0.8687	5.3%	12.2%

C. Computational Complexity

The computational runtime of the New tool - includes a parallelized implementation using OpenMP of the Monte Carlo simulation step from Fig. 1 - is linearly proportional with the number of iterations of the genetic algorithm as shown in Fig. 6. The computational runtime of the New tool is faster by 1.6x an average compared to the Prev1 tool.

IV. CONCLUSION

The main contribution of this paper is the investigation of the impact of multiple levels of uncertainty in design parameters and of uncertainty correlations on the quality of mapping solutions in embedded systems. Simulation results obtained with a custom software tool that solves the problem of mapping with a genetic algorithm approach showed that by not considering uncertainty or by not capturing uncertainty correlations, one can overestimate by up to 16% the performance of the optimal mapping solutions.

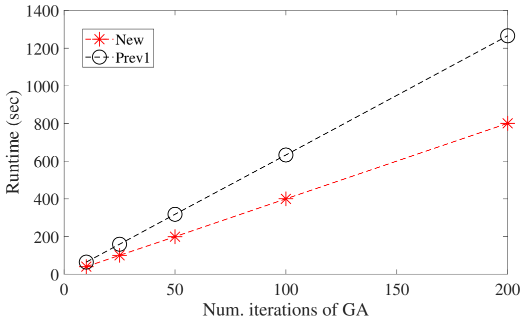

Fig. 6. Computational runtime of the proposed New tool and the *Prev1* tool versus the number of iterations of the NSGA-II genetic algorithm.

ACKNOWLEDGEMENT

This work was supported by the NSF, grant CCF-1524909. Any findings and conclusions expressed herein are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] S.S. Sapatnekar, "Overcoming variations in nanometer-scale technologies," *IEEE JETCAS*, vol. 1, no. 1, pp. 5-18, March 2011.
[2] I. Meedeniya, A. Aleti and L. Grunske, "Architecture-driven reliability optimization with uncertain model parameters," *The Journal of Systems and Software*, vol. 85, no. 10, pp. 2340-2355, 2012.
[3] F. Khosravi, M. Muller and M. Glaß and J. Teich, "Uncertainty-aware reliability analysis and optimization," *ACM/IEEE DATE*, 2015
[4] W. Guan, M.G. Moghaddam, and C. Ababei, "Uncertainty aware mapping of embedded systems for reliability, performance, and energy," *IEEE ISQED*, 2018.
[5] A.D. Pimentel, "The Artemis workbench for system-level performance evaluation of embedded systems," *J. of Systems Architecture*, vol. 3, no. 3, pp. 181-196, 2008.
[6] Cagkan Erbas, "System-Level Modeling and Design Space Exploration for Multiprocessor Embedded System-on-Chip Architectures," *Ph.D. Thesis*, Faculty of Science, University of Amsterdam, 2006.
[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, 2002.
[8] Indika Meedeniya, "Architecture Optimisation of Embedded Systems under Uncertainty in Probabilistic Reliability Evaluation Model Parameters," *Ph.D. Thesis*, Faculty of Information and Communication Technologies, Swinburne University of Technology, 2012.
[9] A.D. Pimentel, C. Erbas, and S. Polstra, "A systematic approach to exploring embedded system architectures at multiple abstraction levels," *IEEE Trans. on Comp.*, 2006.
[10] K. Sigdel, C. Galuzzi, K. Bertels, M. Thompson, and A.D. Pimentel, "Evaluation of runtime task mapping using the Sesame framework," *Int. Journal of Reconfig. Comp.*, vol. 12, pp. 1-17, 2012.
[11] R. Ayari, M. Nikdast, I. Hafnaoui, G. Beltrame, and G. Nicolescu, "HypAp: a hypervolume-based approach for refining the design of embedded systems," *IEEE Emb. Sys. Lett.*, vol. 9, no. 3, 2017.
[12] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: on the design of Pareto-compliant indicators via weighted integration," *Int. Conf. Evol. Multi Criterion Optim.*, 2007.
[13] W. Quan, "Scenario-based run-time adaptive Multi-Processor System-on-Chip," *Ph.D. Thesis*, Faculty of Science, University of Amsterdam, 2015.