JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 07, ISSUE 19, 2020

A METHODOLOGY FRAMEWORK FOR DESIGN SPACE EXPLORATION USING FITNESS PREDICTION TECHNIQUES

C Ramakrishna and Dr. Pramod Pandurang Jadhav

Department of Computer Science & Engineering, Dr. A.P.J. Abdul Kalam University, Indore (M.P.) - 452010

ABSTRACT: Complex software systems have a large number of choices in terms of selection of software components and hardware architectures for implementation. Modern embedded systems are becoming increasingly multifunctional. These design choices create a large space of possible design solutions called the design space. The design process requires exploring through this design space to find valid design solutions before the actual implementation. DSE is the critical design process in which system designs are modeled, evaluated and, eventually, optimized for the various extra-functional system behaviors. This paper presents, a methodology Framework for design space exploration using Fitness Prediction Techniques. The scenario-based DSE uses a multi-objective genetic algorithm (GA) to identifying the mapping with the best average quality. In order to keep the exploration of the scenario-based DSE efficient, fitness prediction is used to obtain the quality of a mapping. This fitness prediction is performed using a representative subset of application scenarios that is obtained using co-exploration of the scenario subset space. Larger subsets will obtain a similar accuracy, but the DSE will require more time to identify promising mappings that meet the requirements of multifunctional embedded systems. Computational tests show that the efficiency of design exploration technique.

KEYWORDS: design space exploration, fitness prediction, subset selection, embedded systems.

I. INTRODUCTION

Designers of modern embedded computer systems face several daunting challenges since these systems typically have to meet a range of stringent, and often conflicting, design requirements [1]. As many embedded systems target mass production and battery-based devices or devices that cannot use active cooling, they should be cheap and power efficient. Mission- and safety-critical embedded computer systems, like those in the avionics and space domains, usually also demand high levels of dependability, which is becoming even more important as the levels of system autonomy rise [2].

Moreover, a great deal of these systems must, increasingly, support multiple applications and standards for which they often need to provide real-time performance. For example, mobile devices must support a variety of different standards for communication and coding of digital contents. In addition, many of these systems also need to provide a high degree of flexibility, allowing them to be easily updated and extended with future applications and standards. This calls for a high degree of programmability of these systems, whereas performance, power-consumption and cost constraints require implementing substantial parts of these systems in dedicated hardware blocks. As a result, modern embedded systems often have heterogeneous multi-processor system architecture. They consist of processors that range from fully programmable cores to fully dedicated hardware blocks for time-critical application tasks [3].

To cope with the design complexity of these systems, the concept of system-level design has been introduced, which raises the abstraction level of the design process.

Design space exploration (DSE) is a key ingredient of system-level design, during which a wide range of design choices are explored, especially during the early design stages. Such early DSE is of paramount importance, as early design choices heavily influence the success or failure of the final product [4].

An important element of system-level DSE is the search for an optimal mapping of the application workload onto the underlying MPSoC platform architecture [5]. Here, the mapping involves two aspects: 1) allocation, and 2) binding. Allocation selects the architectural components for the MPSoC platform architecture (i.e., not all platform components need to be used). Subsequently, the binding specifies which application task or application communication is performed by which multiprocessor system-on-a-chip (MPSoC) component. The number of possible mappings is enormous, especially if there are multiple applications in the workload of the embedded system. During the DSE of embedded systems, multiple optimization objectives – such as performance, power/energy consumption, and cost should be considered simultaneously [6]. This is called multi-objective DSE. Since the objectives are often in conflict, there cannot be a single optimal solution that simultaneously optimizes all objectives. Therefore, optimal decisions need to be taken in the presence of trade-offs between design criteria.

In order to capture the dynamic behavior of multiapplication workloads in system-level design we have introduced scenario-based DSE. An important problem that needs to be solved by such scenario-based DSE is the rapid evaluation of mappings during the search through the MPSoC design space [7]. The number of potential interactions between different applications grows exponentially with the number of applications and application modes that can be simultaneously executed in the embedded system. As a consequence, the potential number of different application scenarios can be huge. Therefore, it is infeasible to rapidly evaluate mappings during the process of early DSE by exhaustively analyzing (e.g., via simulation) all possible workload scenarios. As a solution, fitness prediction [8] can be used to quickly obtain an approximated fitness value. Scenario-based DSE rapidly evaluates mappings for multiapplication workloads during the search through the MPSoC design space. For this rapid evaluation, a co-exploration is performed of the MPSoC design space and the application scenario space. The rest of this paper is organized as follows: Section II relates the literature survey, Section II introduces the Framework for design space exploration, Section IV presents the experiments in which we compare the different fitness prediction techniques and finally paper concludes with Section V.

II. LITERATURE SURVEY

Yong Xie, Gang Zeng, Ryo Kurachi, Hiroaki Takada, and Guoqi Xie, et al. [9] optimize the data throughput of a Controller Area Network bus taking possible attacks on the communication into consideration. They secure the com by adding message authentication codes (MACs) to messages vulnerable to manipulation. The additional overhead induced by these MACs influences the communication delay and system performance. The authors consider the influence of MACs on the performance constraint and describe a method to simultaneously optimize the system's timing behavior and secure the communication. Pimentel, Andy et. al. [10] As inserted systems develop increasingly perplexing and as new applications, for example, IoT require many design limitations, modern

design space exploration techniques are fundamental so as to locate the best tradeoff between various design objectives and their tradeoff. This instructional exercise gives an organized knowledge into the field of design space exploration for inserted systems.

F. Tu, S. Yin, P. Ouyang, S. Tang, L. Liu, and S. Wei, et al. [11] presented a deep neural architecture (DNA), which can reconfigure its data paths to support dataflow techniques for different layer sizes. To find the optimal dataflow pattern, they simulate the analysis of buffer access and DRAM access. This paper performs a simulation by applying the roofline model to analyze the tendencies and provide an intuitive solution for optimal parameters in terms of the number of giga-operations per second (GOPS) and the computation-to-communication ratio (CCR).

S. N. Mahalank, K. B. Malagund and R. M. Banakar et. al. [12] presents a mind boggling Internet of Things framework needs systematic methodology in the underlying phases of the design to freeze between numerous accessible design options. The choices are situated at the framework level, method of information transmission and software module advancement level. The decisions influence the few design 2 objectives demonstrating the choices to speak to a multi-criteria issue to pass judgment on the nature of the new IoT design. A few parts of the design space condition are researched to address the inquiry that emerge at the system level joining stage, specifically the correspondence mode devices, software modules, design reconciliation issue and client administrations are interestingly distinguished from this structure. The necessary interface units, information move mode and software apparatus suite is given utilizing the IoT design space exploration approach. Clients inclinations dependent on the administration prerequisite depicts an end target design reaction that can be utilized in the arrangement model. Nan Feng, Harry Jiannan Wang, and Minqiang Li. et al. [13] propose a technique, which is also based on Bayesian networks for calculating cyber security risk. In their work, the authors describe a security risk management tool which takes into account historical security incidents as well as security expert judgment. Based on these inputs they formulate a risk analysis method. The authors do not consider techniques to mitigate security vulnerabilities, or decrease the security risk.

Kazmierski, Tom & Wang, Leran & Merrett, Geoff & Al-Hashimi, Bashir & Aloufi, Mansour, et. al. [14] presents Fast Design Space Exploration of Vibration-Based Energy Harvesting Wireless Sensors. To research the different exchange offs among these parameters, it is alluring to investigate the multi-dimensional design space rapidly. This paper presents a response surface model (RSM) based technique for quick design space exploration of a total remote sensor hub fueled by a tunable vitality collector A few test situations are considered, which represent how the proposed approach allows the designer to alter a wide scope of system parameters and assess the impact immediately yet with high exactness. In the created toolbox, the assessed CPU time of one RSM estimation is 25 μs and the normal RSM estimation mistake is not exactly 16.5%.

Buchli, Bernhard & Yücel, Mustafa & Lim, Roman & Gsell, Tonio & Beutel, Jan, et. al. [15] remote Sensor Network applications require trustworthy stages that convey right and solid activity over extensive stretches. In any case, application portrayal and accurate system prerequisites speciation can be

muddled because of obscure natural elements and system restrictions. Right now we present another way to deal with design space exploration utilizing a component rich system for encouraged experimentation. We contend that outcomes acquired from experimentation with this stage permit fast speciation of advanced sensor systems at decreased expense.

III. DESIGN SPACE EXPLORATION USING FITNESS PREDICTION TECHNIQUES

A methodology Framework for design space exploration using Fitness Prediction Techniques is represented in below Fig. 1.

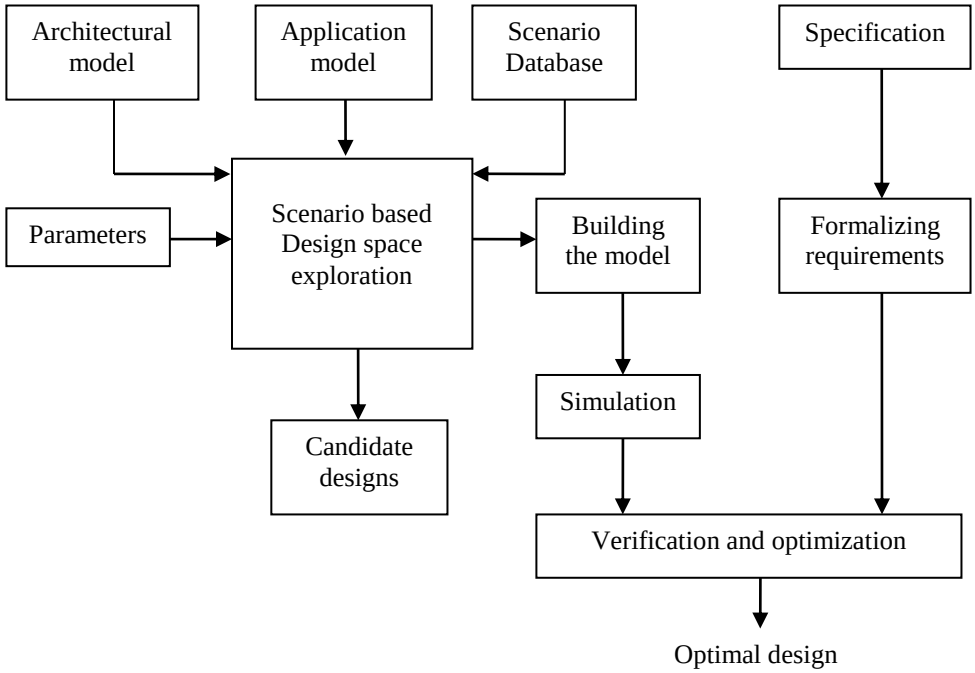

Fig. 1: WORKFLOW OF THE DESIGN SPACE EXPLORATION

The objective of the exploration framework is to provide a static mapping of the multi-application workload onto the MPSoC. This static mapping is used throughout the system's entire lifetime. Therefore, the average behavior of the system for all different possible application workload scenarios must be as good as possible. There are several significant tasks in the design space exploration process, which our methodology is going to solve. First, a designer should create an adequate model of the system using an appropriate tool. Second, a formal specification of requirements should be done, perhaps in parallel with model construction. Finally, he/she needs to run simulations and analyze the provided traces to check whether the requirements are met or not.

The scenario-based DSE explores the mapping of multiapplication workloads onto an MPSoC platform. To this end, a couple of inputs need to be made explicit. Not only the architectural model needs to be given, but also the multiapplication workload. The architectural model describes the complete set of available architectural components (including the available interconnections). This architecture typically will not fit on the

final MPSoC. Therefore, the DSE will reduce this architecture by only using a subset of the architectural resources.

From the multi-application workload two characteristics are required. First, model describes the structure of the applications. Secondly, the possible workload behavior is described explicitly using a scenario database in which the inter- and intra-application scenarios are stored and made explicit. Whereas the input application models specify the structure of each individual (concurrent) application enabling mapping exploration of the application tasks, the scenario database characterizes the different possibilities for multiapplication workload behavior (e.g., which applications or application modes are active at the same time).

A representative subset of application scenarios from the scenario database is used to rapidly evaluate mappings during MPSoC DSE. As the quality of the fitness prediction of a subset is dependent on the mappings that are under evaluation, we have already pointed out that the MPSoC design space needs to be simultaneously co-explored with the application scenario space. The subset selector, tries to identify the best representative subset of application scenarios that is used to evaluate the mappings in the design explorer. The subset selector can be implemented with multiple techniques.

The subset selector is responsible for obtaining a representative subset of scenarios $\widetilde{S_j}$ to predict the fitness of mappings in the design explorer. Due to the potentially large number of scenarios, this selection is not trivial. Ideally, this selection is done statically, before the MPSoC DSE process starts. Therefore, the subset selection needs to be performed dynamically using a training set T_i of application mappings m. The training set T_i is dynamically updated during the complete process of scenario-based DSE.

There is a difference between the real fitness F and the estimated fitness $\widetilde{F_{\widetilde{S_j}}}$. The real fitness uses all possible scenarios to determine the objectives. Therefore, this fitness is independent of the current generation. The estimated fitness $\widetilde{F_{\widetilde{S_j}}}$, however, is only valid during generation j as in the next generation the most representative scenario subset $\widetilde{S_{j+1}}$, which is used to estimate the fitness may be changed.

Before the design explorer can perform any evaluation, the currently representative scenario subset $\widetilde{S_j}$ must be retrieved. Using the obtained subset of application scenarios $\widetilde{S_j}$, the design explorer can quickly evaluate the fitness $\widetilde{F_{\widetilde{S_j}}}$ for all mappings in the current population. As a result, mapping individuals in the parent population may need to be partially reevaluated. Afterwards, the current population is exported to the subset selector. Next, the GA (Genetic Algorithm) can select individuals based on their estimated fitness $\widetilde{F_{\widetilde{S_j}}}$. Similar to the design explorer, a GA is used to identify the best representative subset. A pool of individual subsets follows an evolutionary process with crossover and mutation to obtain a subset of scenarios that is as good as possible

Model checking is a process of automated verification of models against the requirements, which are expressed in some formal language. It examines all possible system scenarios in a systematic manner. Temporal Boolean logic is typically used for requirements specification, though a combination of theories is also possible, for example, difference logic and linear

arithmetic over rationals. For theory combinations, Satisfiability Modulo Theories (SMT) solvers are used. As output, model checking tools provide information whether requirement formulas are satisfied or not. In the latter case, many checkers are able to return a counterexample, which is very useful for debugging and analysis.

A suitable tool for our purposes should either meet these criteria or provide convenient extension possibilities for integrating these components. Another important method used in many design routines is called verification. It aims to ensure the correctness of the model as well as of simulation results. In particular, automated model verification techniques are used for checking simulation traces and also the model itself against the requirements.

The outcome of the DSE is a final set of candidate mappings. This set of mappings is taken from the trainer within the subset selector. Most importantly, the real fitness of these mappings is known. Additionally, the selection procedure of the trainer ensures that the nondominated solutions that are encountered will be kept in the list during the complete design space exploration.

IV. RESULT ANALYSIS

In order to verify the scenario-based DSE, a couple of experiments have been performed. For these experiments, both the multi-application workload and the potential set of architectural components remain fixed. The multi-application workload is generated stochastically with a Python tool based on in such a way that the behavior of ten embedded applications is resembled. The ten applications have a total of 58 processes and 75 communication channels. The multiapplication workload consists of 4607 different application scenarios. We have chosen to use stochastic applications as they provide a wider range of possibilities than real applications do. They can easily be instrumented to mimic real applications, but they also allow for small changes on parameters, such as communication fraction, for a thorough study of the properties of the DSE process. The effect of the subset size on the quality of the identified mapping is analyzed. Therefore, we performed a DSE of 8 hours with the hybrid approach to select a scenario subset. During this DSE, two threads were assigned to the subset selector and six threads were assigned to the design explorer.

1) Accuracy: The larger the subset, the more accurate the fitness prediction in the design explorer is. As a result, it can make better decisions about which individuals must be chosen for the next generation. In general, larger subsets are more likely to be able to identify the fast mapping within a certain time.

2) Overhead: The larger the subset, the longer it takes to obtain the fitness of a single mapping. A longer evaluation time of a single mapping not only means that less individuals can be evaluated in a certain time frame, but also that the GA performs less generations. As a consequence, the search has less benefit from the evolutionary process and it becomes harder to identify good mappings within a certain time frame.

In the experiment, four different subset sizes are used: 0.1%, 1%, 4%, and 16% of the total number of application scenarios. For each individual subset size, the result is averaged over nine DSE runs to take into account the stochastic nature of the GA in the design explorer. Fig. 2 and Fig. 3 shows the results of the experiments.

After a short period (5 min), the evaluation overhead is the most significant effect when

looking at the different subset sizes. For the 1%, 4%, and 16% subsets, the minimal execution time is larger as the size of the used scenario subset increases. In case of the 0.1% subset, this deviation barely decreases over time. This shows that the DSE is far from accurate. For the other subsets, the prediction is accurate enough to result in a very small deviation at the end of 8 h of DSE.

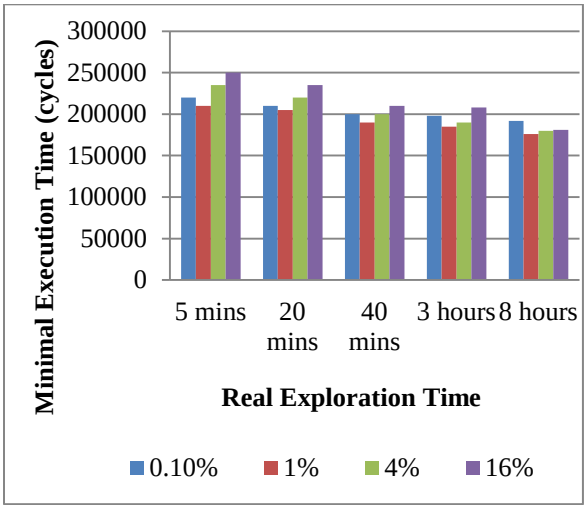

Fig. 2: EFFECT OF THE SUBSET SIZE ON QUALITY PERCEIVED RESULT OF DSE

The overhead effect is not visible any more once the GA is converged. Fig. 3 shows the convergence time (within 1% of final result). In general, a larger subset means a larger convergence time. An exception is the 0.1% subset. The 0.1% subset is not able to provide good mappings as the fitness prediction is not accurate enough. The increased convergence time of the design explorer is also seen in the minimal execution time in Fig. 2. In the first hour, the minimal execution time of the 4% subset larger than the 1% subset. The same holds for the 16% subset. Provided that the subset is accurate enough, the smaller the subset is, the earlier it gets close to the optimal execution time.

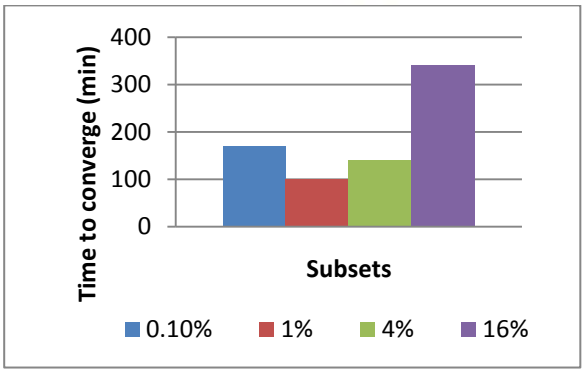

Fig. 3: EFFECT OF THE SUBSET SIZE ON OVERHEAD EFFECT

Therefore, we can speak of an accuracy threshold. Once the accuracy of the subset is above the accuracy threshold, the final GA results are not significantly affected by the subset size. However, due to the overhead effect, the convergence time will increase with a larger subset.

Our final experiment shows a comparison between the different subset selection methods and its effect on the efficiency of the DSE. Therefore, the required exploration time for the scenario-based DSE to identify a satisfying mapping is measured. After all, the faster the DSE can provide results that match the requirement of the user, the better it is. For this purpose, a DSE of 100 min is performed with all the subset selector approaches. Each experiment is performed for three different subset sizes (1%, 4%, and 8%). The results are averaged over nine runs.

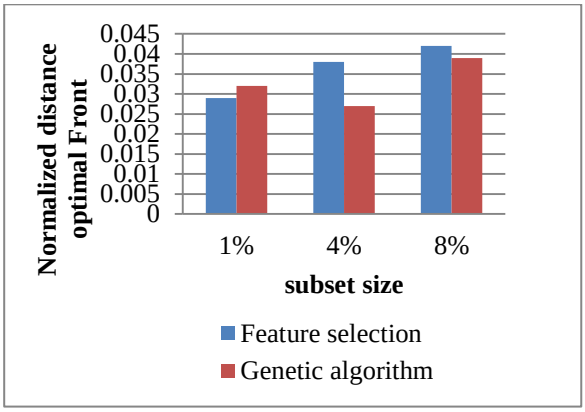

Fig. 4: QUALITY OF DSE FOR DIFFERENT SUBSET SELECTION APPROACHES

We observed that when increasing the subset two effects will occur: 1) higher accuracy, and 2) slower convergence. The FS subset selection has worse results when the subset becomes larger (the smaller the distance, the better). The GA however, shows a somewhat different effect. With 4% it is able to benefit from a subset with a higher accuracy. The slower convergence starts to affect the efficiency from the 8% subset. Comparing the different methods, the GA method has the best results. The only exception is for the 1% subset. In this case the feature selection is still able to search the smaller design space of possible subsets.

V. CONCLUSION

In this paper, methodology Framework for design space exploration using Fitness Prediction Techniques is described. Scenario-based DSE provided an efficient early design space exploration of dynamic multiapplication workloads by co-exploring the design space of multi-application mappings onto an MPSoC with the design space of representative scenario subsets. Fitness prediction is used to obtain the quality of a mapping. This fitness prediction is performed using a representative subset of application scenarios that is obtained using co-exploration of the scenario subset space. The experiments in this paper showed that during the DSE there were two effects: 1) the overhead effect, and 2) the accuracy effect. A larger scenario subset results in a more expensive mapping evaluation. As a result, the DSE required more effort to identify the Pareto front. Additionally, a large subset also meant a higher accuracy. The higher accuracy resulted in a better prediction that made the genetic search to the Pareto front more efficient. The GA however, shows a somewhat different effect. With 4% it is able to benefit from a subset with a higher accuracy. The slower convergence starts to affect the efficiency from the 8% subset. Comparing the different methods, the GA method has the best results.

VI. REFERENCES

[1] Sabeen Masood, Shoab Ahmed Khan, Ali Hassan, "Simulating Synchronization Issues on a Multiprocessor Embedded System for Testing", 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), Year: 2021

[2] Abdulmohsan Aloseel, Hongmei He, Carl Shaw, Muhammad Ali Khan, "Analytical Review of Cybersecurity for Embedded Systems", IEEE Access, Year: 2021

[3] Jonglae Park, Bumgyu Park, Youngtae Lee, Chulmin Jo, Seogjun Lee, "Performance-efficient CPU resource management algorithm on Heterogeneous multi-processor", 2020 IEEE International Conference on Consumer Electronics (ICCE), Year: 2020

[4] Alessio Colucci, Alberto Marchisio, Beatrice Bussolino, Voitech Mrazek, Maurizio Martina, Guido Masera, Muhammad Shafique, "A Fast Design Space Exploration Framework for the Deep Learning Accelerators: Work-in-Progress", 2020 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Year: 2020

[5] Gressl, Lukas & Steger, Christian & Neffe, Ulrich, "Security Driven Design Space Exploration for Embedded Systems," 1-8. 10.1109/FDL.2019.8876944, 2019.

[6] Rajesh Kedia, M. Balakrishnan, Kolin Paul, "Work-in-Progress: A Case for Design Space Exploration of Context-aware Adaptive Embedded Systems", 2019 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Year: 2019

[7] Jan Spieck, Stefan Wildermann, Jürgen Teich, "Run-Time Scenario-Based MPSoC Mapping Reconfiguration Using Machine Learning Models", 2019 ACM/IEEE 1st Workshop on Machine Learning for CAD (MLCAD), Year: 2019

[8] Sam, David & Opoku Agyeman, Michael, "An Overview of Design Space Exploration of Cache Memory," 1-6. 10.1145/3284557.3284558, 2018.

[9] Yong Xie, Gang Zeng, Ryo Kurachi, Hiroaki Takada, and Guoqi Xie, "Security/Timing-aware Design Space Exploration of CAN FD for Automotive Cyber-Physical Systems", IEEE Transactions on Industrial Informatics, PP(8), 2018.

[10] Pimentel, Andy, "Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration," IEEE Design & Test, Vol. No. 1, 10.1109/MDAT.2016.2626445, 2017.

[11] F. Tu, S. Yin, P. Ouyang, S. Tang, L. Liu, and S. Wei, ''Deep convolutional neural network architecture with reconfigurable computation patterns'', IEEE Trans. Very Large Scale Integr (VLSI) Syst., vol. 25, no. 8, pp. 2220–2233, Aug. 2017

[12] S. N. Mahalank, K. B. Malagund and R. M. Banakar, "Design space exploration for IoT based traffic density indication system," International Conference on Recent Trends in Information Technology (ICRTIT), pp. 1-6, Chennai, 2016

[13] Nan Feng, Harry Jiannan Wang, and Minqiang Li. "A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis", Information Sciences, 256, 2014

[14] Kazmierski, Tom & Wang, Leran & Merrett, Geoff & Al-Hashimi, Bashir & Aloufi, Mansour, "Fast Design Space Exploration of Vibration-Based Energy Harvesting Wireless Sensors," Sensors Journal, IEEE. 13. 4393-4401. 10.1109/JSEN.2013.2263792, 2013.

[15] Buchli, Bernhard & Yücel, Mustafa & Lim, Roman & Gsell, Tonio & Beutel, Jan, "Demo Abstract: Feature-Rich Platform for WSN Design Space Exploration," Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN'11. 115-116, 2011