Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2016, 3(11): 30-36

Research Article ISSN: 2394 - 658X

Capturing System Performance Complexity using Simulation Based System Level Modeling

Murigendrayya M Hiremath and A R Aswatha

[1]*Department of Electrical and Electronics Engineering, Jain University, Bengaluru, India*
[2]*Department of Telecommunication Engineering, DSCE, Bengaluru, India*
mmh.eda@gmail.com

ABSTRACT

System level modeling and faster architectural exploration is becoming very essential to achieve a better cost-performance trade off, short time to market, to determine the success and failure of the system and to narrow down the design space early in the design stage [1]. Simulation based system level modeling and analysis involves dynamic and statistical computation of design matrices at different workload events and transitions. This method is faster than RTL (Register Transfer Level) and more accurate than Analytical method [2]. Difficulty in simulation based system level modeling is that the integrating the models of system components at different levels of details with different modeling languages. That leads to increase in learning curve, modeling effort and modeling time. To address this problem, we need to have common platform and method to integrate the analysis specific system component models at different levels of modeling details. Work carried out is to explore the modeling flow for simulation based system level performance modeling, analyzing and optimization using analysis specific generalized parametric models of system components. In this work, system architecture and task mapping is determined based on the analysis and optimization of end to end latency, resource cost and utilization. System components such as multi-core processor, memory, and bus and IO unit can be characterized by generalized standard abstract modeling libraries.

Key words: Performance, Modeling, Simulation, Multi-core

INTRODUCTION

Embedded system designer must of course construct an implementation that fulfils desired functionality, but difficult challenge is to optimize the numerous design metrics [1]. This difficulty is increased by the time to market pressure. Delayed entry to market causes serious negative effect. System level modeling and analysis can be helpful to derive and optimize the performance, power and Cost during early in design phase, is a solution for addressing the design challenges. System level modeling by analytical method is done by solving the series of equations to get the performance and power numbers. This modeling is faster in evaluation time, but less accurate in evaluation numbers and this can be used only for simple system modeling. On the other hand, system modeling at RTL and lower is more accurate but requires detailed system description with more modeling effort and inflexible. Simulation based system level modeling and analysis is the ultimate approach for optimizing the design metrics along with functional implementation. The model can be reconfigured and experimented for different conditions. Component interaction of the model is monitored by simulation called Model of Computation (MOC). Discrete event MOC simulation engine is used in proposed approach, in which system changes instantaneously in response to arrival of discrete events, is being triggered as an event occurs, all concurrent events executed each time and maintains the notation of current time and processes events. Actual electronics system representation is given in equation 1.

$$Digital\ System = Software + Hardware + Inputs/Outputs \quad (1)$$

System model includes behavior flow and mapping models, system traffic, resources and topology, statistical models system objectives, system variables and parameters, model of computation.

$$System\ simulation\ model = model\ of\ computation + system\ variables\ and\ parameters + traffic + behavior\ model + Resources\ model + statistical\ model\ of\ system\ objectives \quad (2)$$

Where

Model of computation	= Discrete event simulator
Behavior model	= Task flow or behavior flow with mapper + Queue + Analysis expressions
Resources model	= Resources (queue with processing time) + Topology
Traffic	= Source and workload for analysis
System Parameters	= Simulation length + Resource Bandwidth + Traffic rate + Experimental condition
Statistical Model	= Performance measures + Other conditions

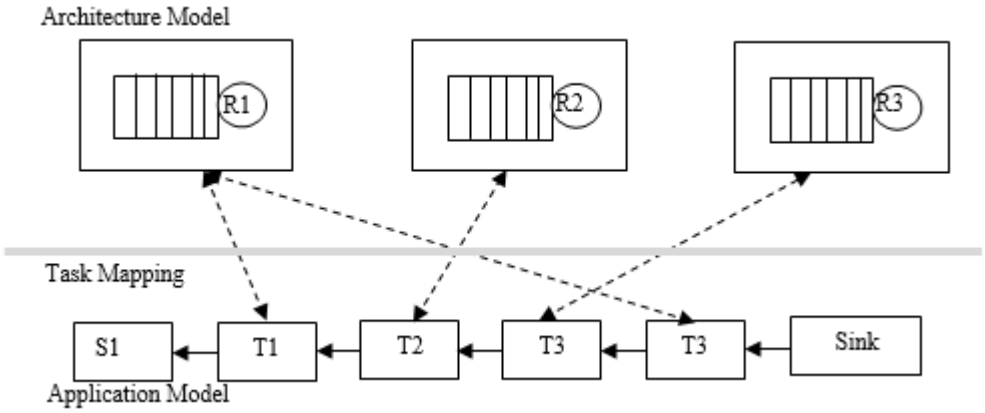


Fig. 1 System Model

Simple System model as shown Fig. 1 consists of application model, architecture model and task mapping. These models are simple enough to understand, are built using abstract library blocks. These library blocks mimic the actual system components specific to system modeling objectives. Equation 2 represents analysis specific system model.

Architecture model are made by scheduler blocks i.e. queue with processing delay representing hardware resources. Application Model represents the application flow, algorithm or task graph. Each task in application model includes the details of resource mapped, and task time. Task time can be relative time or number of cycles per task. Processing time of the resource is can be defined by the following equations. These models can be refined further based on available details.

Processing Time = Relative Task Time of the task = Number of Cycles/ Resource Speed (3)

System modeling represents the working system i.e. system behaviour during applications or software running on system hardware at different input conditions and traffic.

Performance modeling and analysis is used for identifying application scenarios and performance criteria. Performance criteria are system latency, throughput and utilization. Definition of the system latency defined in following way.

System Latency = Time of response at Sink - Time of request at Source

System Latency = Task Processing Time + task Waiting Time

System Latency = Processing Time+ Waiting Time + Memory Access Time + Context Switching + Communication Time + Task Scheduling Time (4)

Resource Utilization is ratio of total processing time of the resource during system execution time to system execution time.

Resource Utilization = processing time of the resource/ system execution time (5)

Simulation Tool used in proposed approach is Visualsim from Mirabilis Design is a system level modeling and analysis tool having built-in simulation engine with pre built analysis specific modeling components such as resource blocks, algorithm flow models, statically blocks and source blocks are available for faster system modeling, analysis and optimization [4]. The rest of the paper is organized as follows. Proposed system level modeling approach is explained in Methods. Experimental results are presented in section results and discussion. Concluding remarks are given in Conclusion section.

METHOD

In Figure 2: Blue shaded block shows pre system modeling steps and purple shaded block shows the steps for developing executable system model. Pre system modeling steps includes documentation of problem definition, real system data collection, details of analysis measures. Developing system level modeling involves the translation of

system specification and analysis objectives into executable system model. The Mentioned steps in the flow are for understanding the concept of system modeling [5]. The sequence of the steps can be varied and be used repeatedly based on refinements, results and levels of system details. Steps involved in simulation based system level modeling and analysis is described as follows.

Problem Identification and Formulation: This step includes problem description, proposed solution, and formulation of the problem, requirements, modeling objectives, expected results, assumptions required, and timeline for modeling.

System Data Collection: In this step System specification, input variables, source of randomness, standard or empirical distributions for simulation are collected.

Develop Parametric Abstract Model and Validate: Develop schematics and flow diagram of the system and translate to simulation model. Verifying simulation model (Trace, Animation, and manually checking results), comparing the model performance under known conditions with performance of real system [6]. Model examined by system experts, analyst Examining model assumptions are correct are done here.

Setup the Configurations and Experimental Condition for Simulation Runs: In this stage Selection of performance measures, input variables and their levels (batch simulation to be considered), list of experimental conditions for runs simulation length, expected output data to be correlated are discussed in this stage. Perform simulation runs: In this stage performance, cost and other metrics numbers are generated in the form of mean, standard deviation, minimum/maximum value.

Collect the Results for Different Simulation Runs and Discuss: Collect results for simulation runs either by batch or individual runs, construct display charts.

Document Model for Communication and Future Use: Document the objectives, assumptions and input variables in detail. Answers for questions with simulation results, and recommendation for further experiments is specified in this stage.

Simulator Selection: Selection of Simulator is based on simulation requirements of the model that is based on type of analysis. Digital simulator engine is used for discrete event simulation.

Declaration and Initialization: This step is as same as the variable declaration in programming language. It includes declaration of parameter, Data structure, global and local memories (variables).

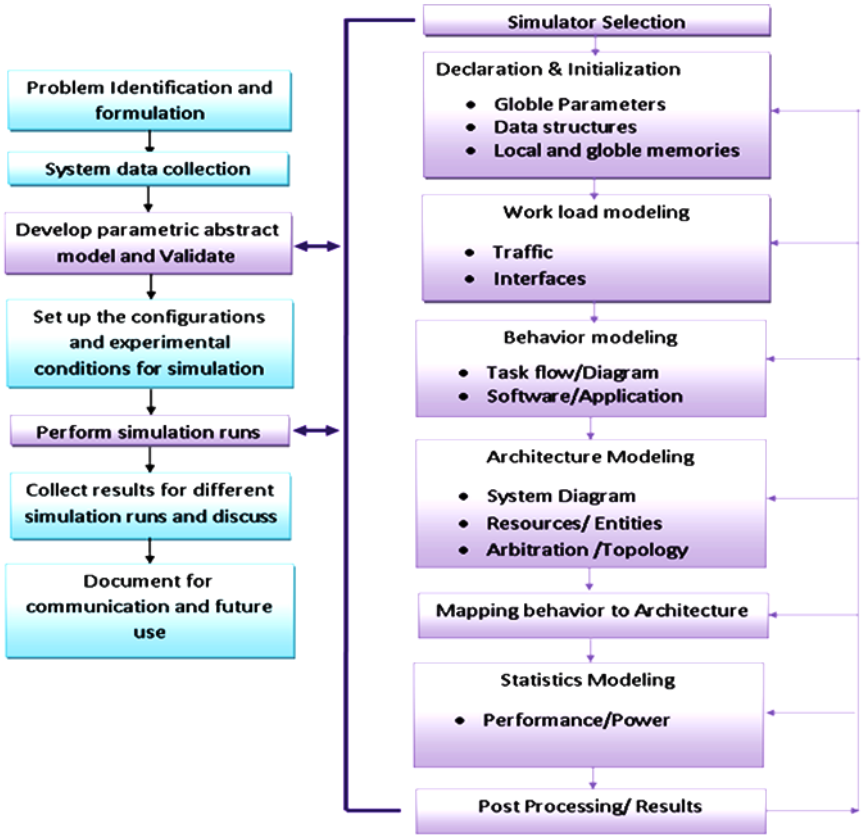


Fig. 2 Simulation based System level modeling flow

Workload Modeling: Workload is to evaluate the effect of different traffic on a system. It represents video frames, signals through pins, network packets and control messages. This can be a distribution based, sequence of reads and writes, and triggers the series of activity. Traffic can be either a data structure, pin value or token (any data type).

Behaviour Modeling: Behaviour flow describes the order and dependency of the tasks that are processed on the data. Behavioural model represents application, algorithm or software flow of the system. It contains timing information for performance modeling.
Architecture Modeling: Architecture represents the computational and communication devices or resources such as processors, buses and memories.

Mapping: Mapping is a process connecting the tasks to the resources; task can be mapped to resources with relative processing time or with the number of instruction cycles or with the list of instruction set of particular task.
Statistics modeling: Statistics models are used to generate the analysis based statistics numbers as given in equation 4 and 5.

Post processing: Collecting the result and plotting for multiple simulation runs for different parameters and their different values is difficult by individual simulation runs. In such case common batch simulation should be considered. Batch simulation is useful for post processing and analysis of simulation results.

RESULTS AND DISCUSSION

Considering a case study consists of three applications and their task flow is shown in figure 3, and configuration details are shown Table 1 and Table2. There are four available resources and their worst case execution time for individual resource at different task is given in the Table 1. Each resource cost is given in Table 2.

Problem Identification and Formulation:

- Develop the simulation model for performance analysis.
- Select the hardware architecture for given application with reasonable latency and cost.
- Conduct the trade study and Optimize the performance of the system.
- Explore the task mapping required to achieve high speed, low cost system.

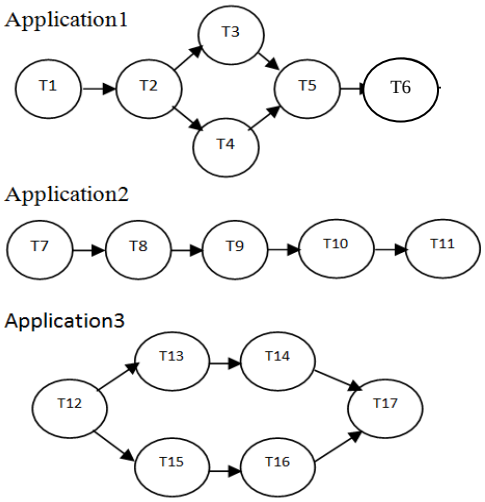


Fig. 3 Task flow diagram

Abstract modeling applications as shown in figure is done by transferring behavior flow of the system to simulating behavioral model by using standard VisualSim library blocks. Mapper block is used to model the task and parameter required for mapper block is task execution time, resource allocation, task priority and task number. Resources are modeled by using scheduler and required parameters are resource speed and scheduling mechanism. Following equations are required for statistical modeling.

$$Latency = TNow - input.Time \quad (7)$$

$$Utilization = Processing\_time / Response\_time \quad (8)$$

To identify the task allocation for optimal latency, allocation parameter of the mapper block is randomized using random function.

Table -1 Cost of Each Resource

	R1	R2	R3	R4
Cost	4	15	3	2

Table -2 Task Worst Case Execution Time for Individual Resource

Task	R1	R2	R3	R4
T1	0.4	0.2	0.3	0.6
T2	0.6	0.3	0.7	0.9
T3	0.4	0.2	0.5	0.6
T4	0.7	0.3	0.8	1.0
T5	0.8	0.4	1.0	1.1
T6	1.2	0.6	1.7	2.1
T7	0.7	0.3	1.0	1.4
T8	1.0	0.5	1.5	1.9
T9	0.6	0.3	0.7	0.9
T10	0.3	0.2	0.4	0.5
T11	0.4	0.2	0.5	0.6
T12	0.5	0.2	0.6	0.7
T13	0.8	0.4	0.9	1.1
T14	0.3	0.2	0.4	0.5
T15	0.4	0.2	0.5	0.6
T16	0.9	0.5	1.0	1.3
T17	0.2	0.1	0.3	0.3

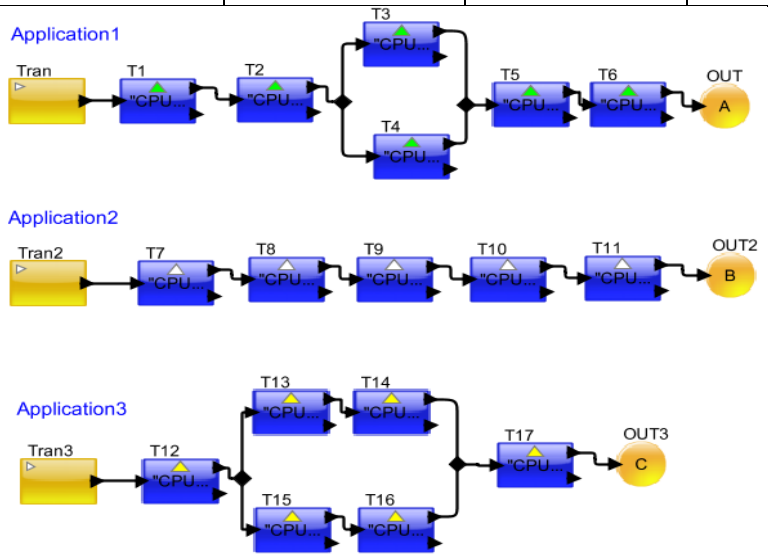
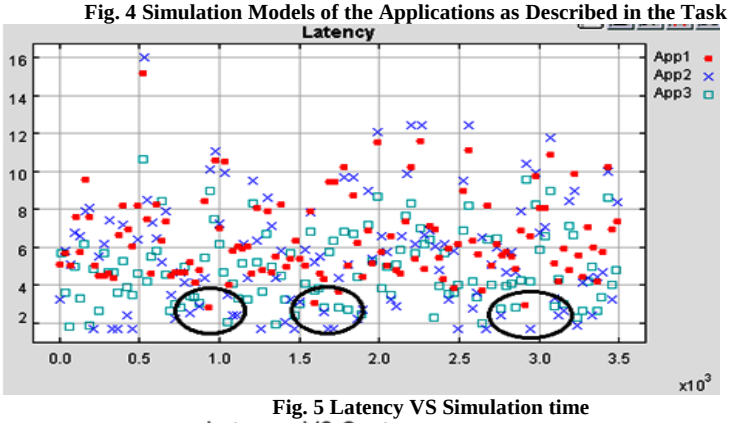
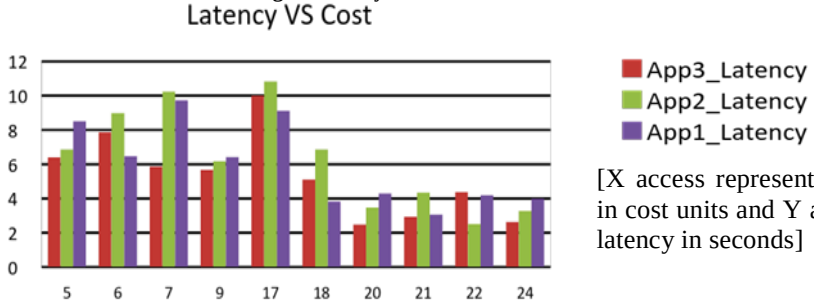


Fig. 4 Simulation Models of the Applications as Described in the Task Flow



[X access represents simulation time in second and Y access represents the latency in seconds]

Fig. 5 Latency VS Simulation time



[X access represents configuration cost in cost units and Y access represents the latency in seconds]

Fig. 6 Latency VS Cost

Collect the Results for Different Simulation Runs and Discuss

All possible latency numbers for three applications shown in figure. From this graph low latency numbers can be identified and experimental setup can be fixed to get the values below 10 units for next simulation run.

Fig. 6 shows the trade-off between latency and cost. Viable solution is identified from graph is cost of 9 units. Resources available for cost 9 units are R1, R3 and R4. Fig. 7 shows the latencies of application 2 and application 3 Vs Application1 for defined architecture. Architecture consists of resource R1, R3 and R4 and cost of resource is 4, 3 and 2 respectively. Lowest latency from this plot for application 1, application 2 and application 3 are 3.4, 5.52 and 5.57 respectively. At this particular latency point utilization of the resources are 85, 97, and 61 and very low. Bottleneck of low resource utilization is identified by timing diagram as shown in Fig. 8. Optimization is done by splitting the task 6 and map to the resource R3 and R4. Optimized latencies three applications are 3.27, 4.37 and 4.89 respectively. Task flow and resources utilization after optimization plots are given in Fig. 9. From Fig. 10 all resources are busy during simulation time and system latency increases gradually because of traffic rate of the workload is faster than the system performance. Due to faster traffic rate number of tasks waiting in the resource queue increases along the simulation time.

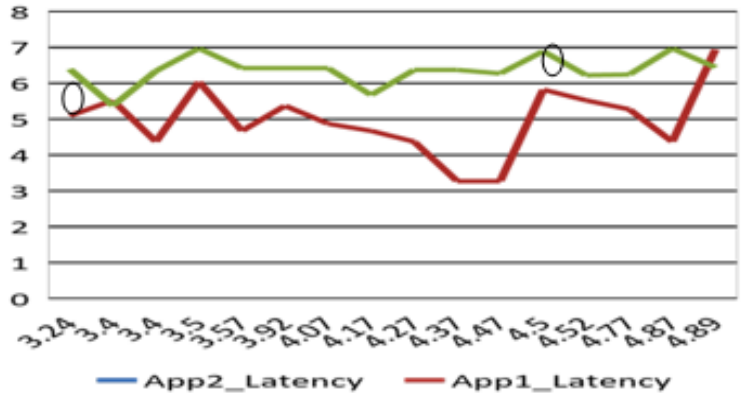
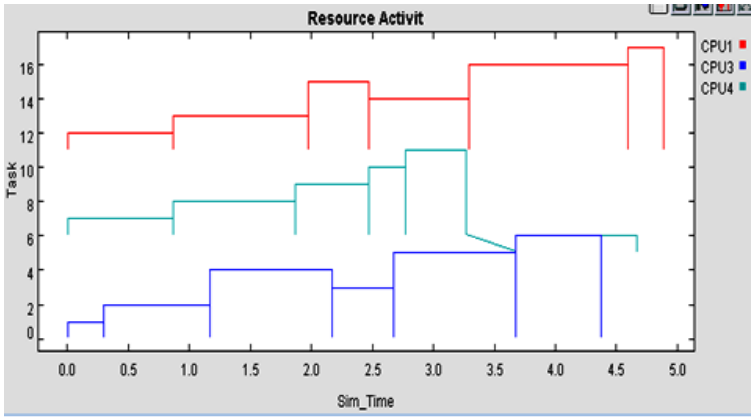


Fig. 7 Latencies of application2 and application3 Vs latency of application1



X access represents simulation time in second and Y access represents task number or taks ID

Fig. 8 Task timing diagram

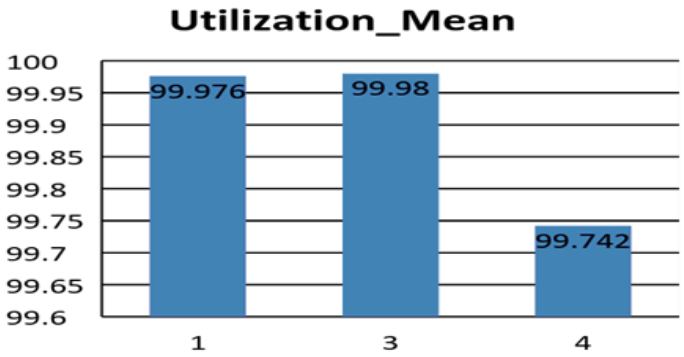


Fig. 9 Resource utilization

X access represents resources R1, R3 and R4 and Y access utilization of each resources

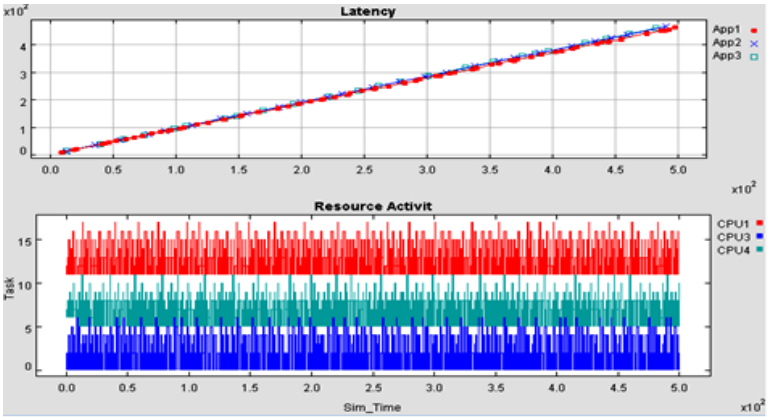


Fig. 10 Resource activity and system latency

CONCLUSION

Concept of simulation based system level modeling and analysis method is presented in this paper by considering an example. Modeling of task flow of multiple applications, resource modeling, and mapping is done using Visul-Sim a system level modeling and analysis tool. Initially all resources are mapped randomly for all tasks and then identified the low latency points at the configuration of reasonable resource cost. The task mapping is further randomized to find out the task mapping at the lowest latency point of defined configuration which is identified in previous simulation runs. Task parallelization is done to achieve maximum resource utilization and low latency. Simulation based system level modeling is faster than RTL simulation and accurate than analytical method and it improves the efficiency of system development in terms performance, time to market, and cost. Performance of the system depends on the waiting time, processing time and accessing time of random arrival of system requests.

Acknowledgements
We thank the authorities of Jain University, Dayananda Sagar College of Engineering and Mirabilis Design Pvt. Ltd for their immense support.

REFERENCES

[1] F Vahid and T Givargis, *Embedded system Design - A Unified Hardware/Software Introduction,* Join Wiley & Sons, New York, **2002.**
[2] Xi Chen, H. Hsieh, F Balarin and Y Watanabe, Logic of Constraints: A Quantitative Performance and Functional Constraint Formalism, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*,**2004**, 23 (8), 1243 – 1255.
[3] Petru Eles, System Design and Methodology/ Embedded Systems Design (Modelling and Design of Embedded Systems) *TDTS30/TDDI08.*
[4] VisualSim : Mirabilis Design, Inc*http://www.mirabilisdesign.com/*
[5] ACJ Kienhnis, *Design space Exploration of stream based Dataflow Architectures: Methods and Tools PhD Thesis,* Delft University of Technology, **1999.**
[6] Anu Maria, Introduction to Modeling and Simulation, Department of Systems Science and Industrial Engineering Binghamton, State University of New York, USA, *Proceedings of the Winter Simulation Conference* ed. S Andradóttir, KJ Healy, DH Withers and BL Nelson, **1997.**
[7] D Chiou, P Jain, L Rudolph and S Devadas, application-Specific Memory Management for Embedded Systems using Software-Controlled Caches, *Design Automation Conference*, **2000**.
[8] Arslan Munir, Ann Gordon-Ross and Sanjay Ranka, Parallelized Benchmark-Driven Performance Evaluation of Symmetric Multiprocessors and Tiled Multicore Architectures for Parallel Embedded Systems, *Wiley-IEEE Press*, **2016**.
[9] AbuAsaduzzaman, Imad Mahgoub, Praveen Sanigepalli, Hari Kalva, Ravi Shankar and Borko Furht, Cache Optimization for Mobile Devices Running Multimedia Applications, *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISME'04)0-769-2217-3/04*, **2004**.
[10] Chandra Prakash Joshi, *A New Performance Evaluation Approach for System Level Design Space Exploration*, PhD Thesis, IIT Delhi, India, **2002**.