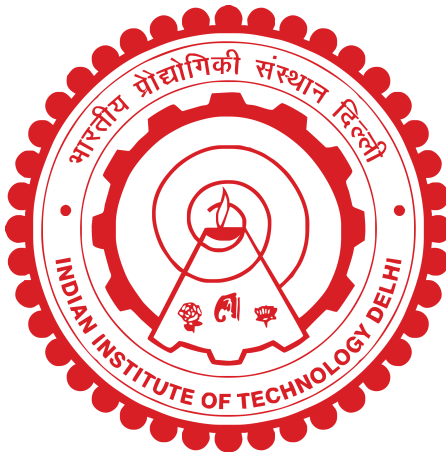


A FRAMEWORK FOR DESIGNING CONTEXT-AWARE ADAPTIVE EMBEDDED SYSTEMS

RAJESH KEDIA



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
OCTOBER 2020**

©Indian Institute of Technology Delhi (IITD), New Delhi, 2020

A FRAMEWORK FOR DESIGNING CONTEXT-AWARE ADAPTIVE EMBEDDED SYSTEMS

by

RAJESH KEDIA

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the



INDIAN INSTITUTE OF TECHNOLOGY DELHI

OCTOBER 2020

DEDICATED TO
My daughter Anika.

Certificate

This is to certify that the thesis titled “**A Framework For Designing Context-Aware Adaptive Embedded Systems**” being submitted by **Rajesh Kedia** for the award of **Doctor of Philosophy** in Computer Science and Engineering is a record of bonafide work carried out by him under my guidance and supervision in the Department of Computer Science and Engineering, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma unless otherwise stated explicitly.

Certain works included in this thesis involved collaboration or use of measurement data obtained by other students, which have been explicitly specified/acknowledged in the corresponding chapters and the part done by those collaborators appeared in their respective reports/theses.

M. Balakrishnan

Professor

Department of Computer Science and Engg.

Indian Institute of Technology Delhi

New Delhi- 110016

Kolin Paul

Professor

Department of Computer Science and Engg.

Indian Institute of Technology Delhi

New Delhi- 110016

Acknowledgments

“I can no other answer make but thanks, and thanks, and ever thanks.”

—William Shakespeare, *Twelfth Night*

I would like to express my sincere thanks and gratitude towards **Prof. M. Balakrishnan**, my Ph.D. supervisor. I interacted with him a few months before joining the Ph.D. programme and he has been advising me in various aspects right through then. Not only has he been a technical mentor, he has provided great insights on many other aspects of an academic life which would be useful for the future. I would also like to thank my co-supervisor **Prof. Kolin Paul** for providing practical insights and inputs on my thesis work. He has always been available for discussions and suggestions and pushed me to write the thesis in time. Both my supervisors have been very supportive during a few hard times in my research and have always motivated me. They actively helped me to complete the thesis during the COVID-19 related lockdown.

Many sincere thanks to **Prof. Preeti Ranjan Panda** for helping me develop additional skills that are necessary for being successful in research and academia. He enabled me to participate in the research paper review process of top conferences and journals which updated me with topics of wider domain. He also provided useful suggestions on my thesis work. **Prof. Anshul Kumar**, my research committee chair, was always approachable and has provided useful feedback on my research. I would also like to thank all instructors who imparted useful skills during my coursework at IITD.

I would like to thank Anupam Sobti for the collaboration on the MAVI project and Shikha Goel for the joint work on DPU. Suman Muralikrishnan provided the data related to the RBD system. Many B.Tech. and M.Tech. students contributed to obtaining data for MAVI. Kapil Dev and Lokesh Siddhu have spent time reviewing and providing useful feedback on various manuscripts related to this thesis. Prasanth V and Kapil Dev have provided useful feedback on the thesis draft.

Going through a long journey of Ph.D. brings various ups and downs, which cannot be sailed smoothly without the support of friends and family. I would like to express my heartfelt thanks to my wife Khushbu and daughter Anika, who have always been with me during various phases of this journey. Lokesh Siddhu has been a very good friend, a technical critic as well as mentor, and provided moral support whenever needed. Not just related to our own research, but our discussions on many other related research topics as well as various social aspects significantly helped my development as a researcher as well as an individual. Many Ph.D. colleagues – Lokesh Siddhu, Sakshi Tiwari, Ankit Anand, Dinesh Khandelwal, Vishal Sharma, Shashank

Sharma, Hameedah Sultan, Divya Praneetha, Hadi Brais, Neetu Jindal, Himanshu Gandhi – have been available for informal interactions and discussions beyond work which was very helpful for rejuvenation. I specifically thank Ankit Anand for involving me into the Ph.D. students cycling group which was truly cherished. Other colleagues – Richa, Iqra, Dilpreet, Ismi, Harsh, Samuel, Solomon – have been helpful in keeping a lively work environment.

Coming back to academia to pursue a Ph.D. after more than 8 years in industry was not an easy decision, specifically with the family-oriented social setup in India. Those who helped me during that time have no lesser contribution than those who helped me during my Ph.D. First, I would thank Ramakrishna Reddy K, my friend and colleague from Texas Instruments (TI), for giving me advice and encouragement on a daily basis. Discussions and inputs from Kapil Dev and Vikas Singal from my B. Tech. class, my brother Akash Kedia, and Prasanth V, C.P. Ravikumar from TI have been very helpful. Rubin Parekhji from TI guided me to make some of the important decisions and provided inputs to set right expectations during a Ph.D. Neeraj Saxena and Shubhra Bhandari enabled a smooth relieving from TI (during the Christmas holidays) in a short notice period. My uncles Hanuman Kedia, Jagdish Kedia, and Patram Bansal have always encouraged me to pursue my interests. My parents, wife, and mother-in-law have always trusted me and been very supportive of my decisions.

On our arrival in Delhi to join IIT, my maternal uncle Patram Bansal and his family kindly hosted us for a month. My younger brother Akash Kedia helped us find a house and settle down in the new city. Our lab and office admins – Ms. Sunita, Ms. Vandana, Ms. Rekha, Mr. Hemant, Mr. Suresh, Mr. Rajesh, Mr. Som Dutt, Ms. Manju, and many others worked behind the scenes in providing logistic support and helped avoid many administrative hassles. They have been very humble and helpful throughout. I would like to thank the Ministry of Electronics and IT (MeitY) for providing assistantship of enhanced value under Visvesvaraya Ph.D. fellowship scheme (MEITY-PHD-2671). I apologize and thank all those who have helped me but their names have been inadvertently missed out.

As we say in Indian tradition, no success comes without the blessings of the elders and the almighty. I am fortunate to have the kind blessings and love of my Late grandfather, Late maternal grandparents, grandmother, father, mother, uncles, aunts, in-laws, siblings, cousins, friends, my entire extended family, and the almighty. I strongly believe that the fortune that came along with the birth of my daughter Anika has helped me secure an admission in a premier institution like IIT Delhi and become the first person in my entire extended family to pursue a doctorate degree. I dedicate this thesis to my daughter.

Rajesh Kedia

Abstract

The rapid advancement in Very Large Scale Integration (VLSI) technology along with Electronic Design Automation (EDA) methodologies has enabled electronic system designers to build increasingly complex and sophisticated systems. Apart from large scale computing, it is impacting almost every aspect of human living ranging from healthcare to education to entertainment. This is being achieved not just through integrating more and more transistors but also by integration of a variety of sensors in a family of devices broadly classified as *embedded systems*. This thesis considers many different aspects of modern embedded systems which together make the process for designing such systems significantly complex. Firstly, due to their recent success, machine learning, natural language processing, and computer vision techniques are increasingly becoming an integral part of embedded systems. A key characteristic of algorithms in these domains is that they are not “perfect” – the results generated have a certain accuracy and thus implicitly a certain error rate. The same task may be implemented with many choices which differ in accuracy and computation efforts and hence provide a few trade-off options to system designers. Interestingly, even these accuracy numbers may change with the nature of data, thereby enlarging the design space even further with many “modes” for the same tasks. Secondly, today’s platforms provide multiple compute fabrics ranging from CPU, GPU, FPGA to customised accelerators in the same device. The presence of multiple fabrics very often increases options for execution of application tasks and therefore, provides additional flexibility but again with trade-off among parameters like performance, energy consumption, etc. Thirdly, the systems need to adapt the operating task modes in response to changes in various factors that are external to the system. We refer to these factors together as *context*. We call systems that are being designed to address the above three aspects as *Context-aware Adaptive Embedded Systems (CAES)*.

Consider early stages of designing such systems when even platform definition is still to be decided. The system designer needs to evaluate various options of task modes implemented along with the possible choices for platform components and the range of context in which the device is expected to operate. Clearly, there is a need for joint consideration of platforms, tasks,

and context at this stage to reach good solutions. On the other hand, such a joint exploration results in many-fold increase in the possible space of implementation, thereby significantly increasing the complexity of designing a CAES. Addressing such complexity through *Design Space Exploration (DSE)* in a structured manner is a key contribution of this thesis and forms the first step of the proposed two-step process for designing efficient CAES. In this step, a constraint based formulation of the DSE problem along with an iterative pruning approach is used to explore the design space. Various enhancements like quantization, genetic algorithm, and clustering are also integrated into the flow. Further, the DSE step uses a system model as an input, for which a *graphical representation* is proposed in the thesis. This representation is capable of capturing different choices of different task modes and components as well as the effect of context. Various rules are defined to automatically translate this representation into constructs from constraint logic programming (CLP) to enable its direct use in DSE.

The second step of the proposed CAES design process uses the reports generated from the DSE step to automatically identify suitable mapping of various contexts to task modes (called *run-time controller specification*), corresponding to the platform chosen during the DSE step. The flow uses concepts from multi-valued logic minimization to further minimize the generated run-time controller specification in order to reduce the total number of mappings. Such minimization process also provides insights into the operation of the system, specifically sensitivity to different context parameters. These insights can be useful for further refinements in the implementation.

Three different real systems have been used as case studies to demonstrate the applicability of the proposed design process. These case studies are diverse in their end-application, in the nature of the platforms and task modes, and in the metrics and context that are relevant for these systems. For all the three systems, many new design options become feasible due to consideration of components, task modes, and context together. The thesis presents such design options as well as associated trade-offs identified using the proposed framework.

Overall, this thesis defines a new class of systems named CAES and presents a comprehensive design methodology for such systems which can be used from very early stage in the design process.

ý ò â ý ò l , ,
- l l - l ,
ñ ¼ ò ý ö ¼ ò ()
l l

ñ ò l (-
) ò â
ã - ý
ã
ò ò l l ,
l ð

é l l l
- â ö l l ô
- , ý
, l l
ý - -
, l
ÿ

Contents

Certificate	ii
Acknowledgments	iii
Abstract	v
List of Figures	xii
List of Tables	xiii
1 Introduction	1
1.1 Emerging Embedded Systems	1
1.1.1 Complexity of Tasks/Application	1
1.1.2 Complexity of Embedded Platforms	3
1.1.3 Context-awareness in Embedded Systems	4
1.2 Examples of Context-aware Adaptive Embedded Systems	5
1.2.1 Traffic Monitoring System	6
1.2.2 Video Streaming Application	6
1.2.3 Mobility Assistant for Visually Impaired	7
1.3 Terminology	8
1.4 Motivational Example and Illustration	10
1.5 Summary and Outline of the Thesis	12
2 Background and Related Work	15
2.1 System Design and Design Space Exploration	16
2.1.1 Exploring Task Modes and Resources Choices	18
2.1.2 Tools and Frameworks for System Design	23
2.2 Context-aware and Scenario-aware Systems	25
2.3 Summary	26

3	Overall Design Flow and System Modeling	29
3.1	Overall Design Flow	30
3.2	System Model	31
3.2.1	Metrics for Evaluating Design Points	31
3.2.2	Task Model	32
3.2.3	Resource Model	33
3.2.4	System Level Interactions	34
3.2.5	Pruning Rules and Designer Specified Constraints	35
3.2.6	Illustration of the Model Using an Example	37
3.3	Specifying the System Using Constraint Logic Programming	40
3.3.1	CLP Facts	40
3.3.2	CLP Rules	41
3.4	Summary	42
4	Design Space Exploration	43
4.1	Introduction	43
4.2	Related Work	44
4.3	The Proposed DSE Framework	45
4.3.1	Pruning Strategy	46
4.3.2	Decision Making	55
4.4	Summary	56
5	Graphical Representation of Design Space	57
5.1	Introduction	57
5.2	Related Work	58
5.2.1	Design Space Representation/Model for Embedded Systems	58
5.2.2	Variability Modeling	59
5.3	Proposed Representation	61
5.3.1	Basic Constructs of UCM	61
5.3.2	Application Part of UCM	63
5.3.3	Resource Part of UCM	66
5.3.4	Capturing Constraints and Context	67
5.4	Conversion of UCM to CLP	68
5.4.1	Generating CLP Program	68
5.4.2	Overall Conversion Algorithm	71
5.5	Summary	73

6	Run-time Controller Specification Generation	75
6.1	Introduction	75
6.1.1	Motivation	76
6.2	Background and Related Work	77
6.2.1	Controller Specification Generation/Synthesis	78
6.2.2	Controller Specification Minimization/Optimization	78
6.2.3	Overview of Quine-McCluskey Method	79
6.3	Proposed Approach	80
6.3.1	Generate Context to Mode Mapping (Step-1)	81
6.3.2	Controller Minimization (Step-2)	81
6.3.3	Scalability Study	87
6.4	Summary	89
7	Case Studies	91
7.1	Refreshable Braille Display (RBD)	92
7.1.1	RBD: System Description	92
7.1.2	RBD: System Modeling	93
7.1.3	RBD: Design Space Exploration	94
7.2	Deep Learning Processor Unit (DPU) based Driver Assistance System	97
7.2.1	DPU: Background and System Description	97
7.2.2	System Model for a DPU Based System	98
7.2.3	Design Space Exploration for a DPU based System	103
7.3	MAVI: Mobility Assistant for Visually Impaired (MAVI)	112
7.3.1	MAVI: System Description	112
7.3.2	MAVI: System Modeling	113
7.3.3	MAVI: Graphical Representation	114
7.3.4	MAVI: Design Space Exploration	116
7.3.5	MAVI: Run-time Controller Specification Generation	120
7.4	Summary	122
8	Conclusion and Future Directions	125
8.1	Conclusion	125
8.2	Future Research Directions	127
I	List of Acronyms	129
II	Pruning Rules for DPU: Illustration	131

III Detailed Acknowledgements for Collaborative Works	133
Bibliography	135
List of Publications	147
Biography	149

List of Figures

1.1	Complexity of embedded systems in terms of platform, application, and context	2
2.1	A typical design process for embedded system design	16
2.2	Various aspects in DSE: prior works landscape	17
2.3	Categorizing various prior works based on their DSE scope	20
3.1	Proposed two-step process for designing a CAES	30
4.1	High level view of the proposed DSE framework	46
4.2	A snapshot of our visualization tool highlighting its features	48
4.3	Visualizing the design points when quantized with different granularity	50
4.4	Chromosome structure for use in genetic algorithm for DSE	52
4.5	Comparing CLP based and genetic algorithm based DSE	53
4.6	Variation in the number of clusters formed with allowed distance	54
5.1	Node types in a UCM model	61
5.2	Connecting application and resource parts in a UCM model	63
5.3	An example annotation of resource requirements for a task	64
5.4	Application part of UCM for the example system	65
5.5	Resource part of UCM model for the example system	67
5.6	Capturing constraints and context in a UCM model	68
6.1	Mapping of different external inputs to system modes for an example system	76
6.2	Run-time controller specification generation steps	82
6.3	Comparison of grouping for QMM used in prior works and proposed approach	84
6.4	Example to illustrate the minimization step in run-time controller specification generation	85
6.5	Reduction in the number of comparisons due to proposed grouping for QMM	87
6.6	Size of table T3 versus size of table T2 when increasing the ECI granularity	88

7.1	System level block diagram of RBD system	92
7.2	A Braille module with 10 characters	92
7.3	Histogram of error in the estimation of battery life for the RBD system	94
7.4	Battery life for different reading speeds and ambient temperature for RBD	95
7.5	Battery life (for B3) for various number of Braille modules and reading speeds	96
7.6	Number of text characters that can be read with a single charging of the battery for an RBD	97
7.7	Mapping of tasks on FPGA for different choices of DNN and DPU	98
7.8	System level view of DPU and memory interface	100
7.9	Increase in execution time due to memory contention caused by different DNNs executing concurrently on 3 DPUs	100
7.10	Error in estimating the interference for various DPU sizes	102
7.11	Energy consumption for different DPU sizes for various DNNs	104
7.12	Reduction in resource count and total power of FPGA chip for different DPU sizes	105
7.13	Energy consumption for different number of DPUs (B4096) for various DNNs	106
7.14	Accuracy for various vehicle speeds and FPGA types for DAS	110
7.15	Pareto points obtained after DSE for DAS	111
7.16	Exploration time comparison for different number of tasks for a DPU based system	111
7.17	Primary tasks and resources in the MAVI system	112
7.18	Dependencies between various AP metrics, platform, and context for MAVI	113
7.19	UCM application model showing the choices in task modes for MAVI system	115
7.20	UCM resource model showing the resource choices for MAVI system	115
7.21	Chromosome structure for use in genetic algorithm for MAVI	116
7.22	Hypervolume variation for MAVI for various parameters of genetic algorithm	117
7.23	Battery life versus detection accuracy for different walking speeds for MAVI	117
7.24	Battery life with different battery models for different walking speeds for MAVI	118
7.25	Cost for various battery models and cores for MAVI	118
7.26	Detection accuracy (max.) versus WS for different number of CPU cores and presence of FPGA for MAVI	119
7.27	Controller specification for different platforms for the MAVI system	121

List of Tables

1.1	Tasks and modes for the example system	11
1.2	Feasible design options for the example system	11
2.1	Approaches for assessment of design points with associated objective metrics .	19
2.2	Summary of prior works on DSE: scope and limitations	23
2.3	Scope of related prior works vis-a-vis CAES	27
4.1	Summary of prior DSE works using constraint based pruning approach	45
4.2	Number of distinct design points for various quantization granularity for MAVI case study	51
5.1	Summary of prior works in representing design choices	60
5.2	Characteristics for resource and application parts	66
6.1	Effect of number of ECIs on size of tables T2 and T3	89
7.1	A few choices of operational specification for RBD	95
7.2	Characteristics of various DNNs used in our experiments	101
7.3	DNN accuracy and required frame rate for DAS	109
7.4	Various design choices for DAS	109
7.5	A few choices for operational specification for MAVI	120
7.6	Minimized controller specification for the MAVI system with 4-cores	122
I.1	List of acronyms used in the thesis	129
II.1	Accuracy of tasks for different DNNs	131
II.2	Execution time of DNNs on different DPUs	131