The virtuous circle between electronic systems and their design tools

Ever since the nomadic age, humans have known the concept of 'more' or 'better'. Simple questions such as 'wouldn't it be better if I don't have to hunt every day', or 'I wish I had more to eat' perhaps led to agricultural cultivation and storage of food grains. It was this quest of humans which led to the invention of fire and wheel. This insatiable desire of humans to do more or better is still enabling newer developments and has revolutionized many different fields including electronics. What yesterday was deemed impossible for an electronic system is a reality today. Each one of us has witnessed the burgeoning electronic era in the last two decades, with a significant transformation in personal space and the society due to extensive penetration of mobile phones and personal electronic gadgets. Despite such advancements, the quest for humans to always improve things at hand is making the electronic systems further better by integrating more features into a constantly decreasing form factor. Constant developments in chip manufacturing technology as well as electronic design automation (EDA) are key drivers to such advancements. EDA entails designing an electronic system with the help of software tools on a computer and enables engineers to develop complex electronic systems which are otherwise almost impossible to design manually. The ever increasing growth in the complexity of electronic systems is 'bootstrapping' the capabilities of EDA.

While the complexity of the **hardware platforms** has been ever-growing due to advancements in chip manufacturing technology, the **complexity of applications** executing on recent electronic systems has grown manifolds. Many recent and emerging applications are based on machine learning, artificial intelligence, or computer vision. Since they try to mimic human perception which is not always perfect, imperfect results (subject to certain limits) are acceptable for such applications in many scenarios. Presence of such **tolerance to errors** balances the accuracy/quality of the results and the amount of time or hardware resources required for processing the result. As an example, in an image with 5 faces, locating all the 5 faces accurately is ideal for a face detection application, but might take almost double the time compared to correctly tagging 4 faces and leaving one difficult match unanswered. With the ubiquity of mobile phones and other portable gadgets, there are numerous applications that can tolerate such imperfections. Such flexibility to deliberately introduce slight imperfections is a new enabler for the never ending quest of designers to build electronic systems that are smaller, faster, and consume lesser energy.

Along with the growth in complexity of hardware and applications, **context awareness** has been another game changer for the electronic systems. An electronic system gains the basic smartness by being aware of the context (external conditions) around it. Automatic brightness control of the screen based on lighting conditions, switching to hands-free mode while driving a car, etc. are simple examples of context awareness present in today's systems. A highly sophisticated support for context-awareness would be a key factor for the success of emerging complex systems like autonomous vehicles and personalized medical devices.

Such developments and trends in electronic systems have significantly increased the **complexity of building** future systems. In such a situation, a typical conversation between an electronic system and EDA software goes like this:

- ➔ *Electronic systems (ES):* Hey, I am very happy with the improvements experienced over years. Many thanks to you for enabling such a development.
- ➔ *EDA:* That's nice to hear. We developed together as I also experienced several improvements in order to improve your capabilities. How do you see the future though?
- ➔ *ES:* With various choices of accuracy levels, context conditions, and hardware options getting combined together in recent systems, the future seems challenging for both of us.
- ➔ *EDA:* Yes, these choices form a vast 'design space' and a systematic 'design space exploration' (DSE) appears very complex. But, such challenges have always been fuelling our capabilities. So, I am hopeful.
- ➔ *ES:* Oh, ya! Earlier when I was simple, you were also simple and provided minimum support to designers; today when you have become highly sophisticated, my complexity has also grown considerably, requiring further support from you.
- ➔ *EDA:* True. We form a perfect virtuous circle. Let's together chase the limit of sophistication!

This research focuses on an efficient methodology for designing application specific electronic systems (commonly known as embedded systems), by considering an amalgamation of the three variants of complexity discussed so far - complex hardware platforms, newer applications with tolerance to imperfections, and context awareness. The conventional approach of building electronic systems considers a fixed specification and builds a system to meet the specification. However, with accuracy and context being available as tunable knobs to the designer, various hardware aspects like execution delay, energy consumption, etc. must be considered while defining the specification. A bigger and faster processor could support higher accuracy but would incur higher cost and energy consumption. A system with higher energy consumption could use a bigger battery to elongate the battery life but would make the product bulky. Such **inter-dependencies** between system specifications (accuracy, battery life, weight, or cost) and hardware choices (processor, battery, or approximation level) motivate us to evaluate and analyze design choices together with system specifications. We use **Constraint Logic Programming (CLP)**, a highly efficient constraint based pruning framework, to eliminate design choices of lesser interest. The behavior of various applications, hardware components, and their interactions are modelled into CLP format. We also add the relationship of various context inputs and specify various tentative specification requirements as constraints into the CLP model. A CLP solver uses this model to return only the choices that satisfy the specified constraints. The designer can subsequently relax the constraints if the feasible choices are not of interest; or could tighten the constraints if a lot of design choices are shown as feasible. Such an iterative process can help eliminate unwanted choices and enable the designer to focus on choices of interest.

To facilitate interaction of the designer with the CLP solver, this research proposed and developed a **visualization tool** which can be used to control and observe various iterations of design space exploration. Such a tool can interactively plot various tradeoffs in accuracy, battery life, cost, weight, etc. using various kinds of 2D and 3D plots. It can also update the constraints using a text box or slider control and regenerate the newer plots according to the updated constraints by interacting with the CLP solver in the backend. Our aim is to enable non-expert designers to design complex systems for which a **graphical representation** is proposed to automatically generate the CLP model from a graphical model captured by the designer, without worrying about the syntax. Such a framework consisting of a graphical representation, constraint based elimination of unwanted design choices, and a visualization tool can help advance the EDA tools to tackle the growing complexity of designing emerging embedded systems.

Many real-life portable systems depict the described complexity of electronic systems. We considered data from **three real systems** to deploy our proposed framework into their design process. The first system is a Refreshable Braille Display (RBD) used to render electronic text as Braille script on a portable electronic device. The second system, Mobility Assistant for Visually Impaired (MAVI), is a camera based personal mobility assistant for visually impaired users to warn them of potential obstacles like stray animals or objects of interest like signboards in the walking path of the user. The third system is a driver assistant system (DAS) implemented using specialized hardware units to detect objects of interest and assist the driver by identifying people or vehicles in the driving path. Deploying the proposed framework on these systems identified many new promising design choices which could not be obtained if specifications and design were considered independently. The proposed methodology could successfully evaluate these systems which significantly differ in their features, implementation, and metrics of interest, thereby demonstrating the **versatility of the methodology**. However, this is just a beginning and a small attempt towards making the EDA methodologies scale to support emerging complex systems. Recently, COVID-19 has created an unprecedented demand which has transformed many electronic systems and softwares such as contact tracing apps, video-conferencing solutions, remote classrooms, etc. With such newer requirements emerging and the insatiable nature of human beings to always have better, I believe that the virtuous circle between EDA and electronic systems would continue making each other more sophisticated than ever before.