Virtual Platforms for Energy Efficient Execution of Mixed Time-Criticality Applications

Anca Molnos[1], Sander Stuijk[2], Benny Akesson[2], Andrew Nelson[1], Sjoerd te Pas[2], Martijn Koedam[2], Radu Stefan[2], Ashkan Beyranvand Nejad[1], and Kees Goossens[2]

[1] *Delft University of Technology, the Netherlands,* [2] *Eindhoven University of Technology, the Netherlands.*

System-on-Chip (SoC) design becomes increasingly complex, as a growing number of applications are integrated in such systems. These applications have mixed time-criticality, i.e., some have firm-, some soft-, and some non-real-time requirements. Executing mixed time-criticality applications on a complex SoC in an energy-efficient manner leads to the following problems.

First, to reduce cost, platform resources, e.g., processors, interconnect, memories, are shared between applications. However, sharing causes interference between applications, making their behaviors interdependent. This results in two problems for SoC design and verification: 1) accurate system-level simulation and several approaches to formal verification are not applicable, because of the explosion in the number of possible combinations of applications, inputs, and resource states and 2) verification becomes a circular process that must be repeated if an application is added, removed, or modified, making integration and verification dominant parts of SoC development, in terms of time and money.

Second, energy management is traditionally regarded as a system-level issue. Management policies are hence generally developed for entire SoCs. Applications with different time-criticality inherently have different requirements, hence utilizing a single policy for an entire SoC leads to energy-inefficient execution, or even violation of real-time constraints.

The CoMPSoC platform addresses these problems by executing each application in an independent virtual platform. The virtual platforms are *composable*, i.e., cannot affect each other's behaviors. In the temporal domain an application's actual execution never varies by even a single clock cycle. Similarly, the energy and power behaviors of applications are composable. As a result, applications can be designed, developed, verified, and executed in isolation.

Composable virtualization alleviates the verification problem in the mixed-time criticality domain in two ways: 1) verification becomes a non-circular process, and 2) the time required by simulation-based verification is reduced, since only a single application in its virtual platform has to be simulated.

Furthermore, energy usage can be managed per-application. Each application receives an independent energy and/or power budget, which it can manage with its own application-specific policy. Designers can optimize their application's energy consumption using their virtual platform's specification, independently of the development of other applications for the same platform.

The virtual platforms are also *predictable*, meaning that all interference is bounded. This makes them virtualized also in terms of performance bounds, which enables firm real-time applications to be verified using formal performance analysis frameworks. Currently, we are utilizing the SDF3 design-flow to automatically analyze firm real-time applications, and map them on a virtual platform. Furthermore, CoMPSoC provides a uniform execution model for several computation models, namely, dataflow, Kahn Process Networks, and simple threads, to suit each of the firm-, soft-, and no real-time domains.

Our University Booth presents two demonstrators. First, we demonstrate the SDF3 design-flow that maps multiple multimedia applications on an FPGA instance of the CoMPSoC platform, while meeting their timing constraints. Second, on the same instance of CoMPSoC, we demonstrate per-application Quality-of-Service and energy management for two H263 applications. Our demonstrators also show that applications execute composably, without interfering with each other in time or in energy consumption.