Embedded System-Level Platform Synthesis and Application Mapping – ESPAM: Design Flow Overview

Todor Stefanov Hristo Nikolov Ed Deprettere
Leiden Embedded Research Center
Leiden Institute of Advanced Computer Science
Leiden University, The Netherlands
{nikolov,stefanov,edd}@liacs.nl

ESPAM is a tool for automated multiprocessor system design and implementation. ESPAM moves the design specification from RTL to a higher, so called System-level of abstraction. Starting from system level platform and application specifications, a multiprocessor platform is automatically synthesized and the application is systematically mapped onto it in an automated way. Our system design methodology is depicted as a design flow in Figure 1. There are three levels of specification in the flow. They are SYSTEM-LEVEL specification, RTL-LEVEL specification, and GATE-LEVEL specification. The SYSTEM-LEVEL specification consists of three parts:

- *Platform Specification* describing the topology of a platform using our system level platform model, i.e., using generic parameterized system components. The components are grouped into 1) *Processing Components*, 2) *Memory Components*, 3) *Communication Components*, 4) *Auxiliary Components*. Using the platform model a system designer can specify many alternative platform instances easily.
- *Application Specification* describing an application as a Kahn Process Network (KPN). A KPN specification exposes task-level parallelism available in an application and makes the data communication between tasks explicit. A KPN is a network of concurrent autonomous processes that communicate data in a point-to-point fashion over unbounded FIFO channels, using a blocking-read synchronization primitive.
- *Mapping Specification* describing the relation between all processes and FIFO channels in *Application Specification* and all components in *Platform Specification*. By changing the *Platform* and *Mapping Specifications* different alternative implementations for an application can be obtained easily.

The SYSTEM-LEVEL specification is given as input to ESPAM. First, ESPAM constructs a platform instance following the platform specification and runs a consistency check on that instance. The platform instance is an abstract model of a multiprocessor platform because at this stage no information about the target physical platform is taken into account. The model defines only the key system components of the platform and their attributes. Second, ESPAM refines the abstract platform model to an elaborate (detailed) parameterized RTL model which is ready for an implementation on a target physical platform. We call this refinement process platform synthesis. The refined system components are instantiated by setting their parameters based on the target physical platform features. Finally, ESPAM creates program code for each processor in the multiprocessor platform in accordance with the application and mapping specifications.

The output of ESPAM, namely a RTL-LEVEL specification of a multiprocessor system is a model that can adequately abstract and exploit the key features of a target physical platform at the register transfer level. It consists of four parts:

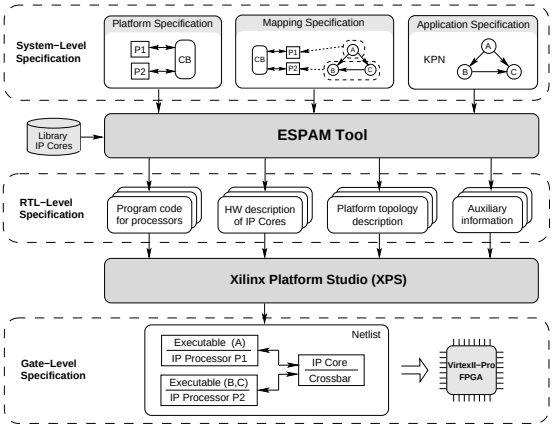

Figure 1: ESPAM System Design Flow.

- *Platform topology* description defining in great detail the processors network (multiprocessor platform).
- *Hardware descriptions of IP cores* containing predefined and custom IP cores used in 1). ESPAM selects predefined IP cores (processors, memories, etc.) from *Library IP Cores*, see Figure 1. Also, it generates custom IP cores needed as a glue/interface logic between components in the platform.
- *Program code for processors* — to execute the application on the synthesized multiprocessor platform, ESPAM generates program source code files for each processor in the platform.
- *Auxiliary information* containing files which give tight control on the overall specifications, such as defining precise timing requirements and prioritizing signal constraints.

With the descriptions above, a commercial synthesizer can convert a *RTL-Level* specification to a *Gate-Level* specification, thereby generating the target platform gate-level netlist, see the bottom part of Figure 1. This *Gate-Level* specification is actually the system implementation. The current prototype version of ESPAM facilitates automated multiprocessor platform synthesis and application mapping using Xilinx VirtexII-Pro FPGAs. ESPAM uses the Xilinx Platform Studio (XPS) tool as a back-end to generate the final bit-stream file that configures a specific FPGA. We use the FPGA platform technology for prototyping purposes only. Our ESPAM is general and flexible enough to be targeted to other physical platform technologies. A real-life industrially-relevant application, namely Motion-JPEG encoder, has been fully implemented by using the ESPAM and XPS design tools. A simple design space exploration has been conducted for Motion-JPEG multiprocessor systems featuring up to 8 MicroBlaze processors connected through a crossbar switch or in a point-to-point network. The exploration is based on real multiprocessor system implementations generated by our ESPAM tool. Therefore, for all these multiprocessor systems we have HW/SW demos.