We introduce a high-level programming language dedicated to the implementation of stream-processing applications on reconfigurable hardware. The language is based upon the actor/dataflow model of computation: an application is described as a network of autonomous processing elements (actors) exchanging tokens through unidirectional channels (FIFOs).

The behavior of actors is specified using a language inspired from the HUME programming language [1]: it is described as a set of generalized transition rules, where each rule consists of a set of patterns, involving inputs and possibly local variables, and a set of expressions, describing modifications of outputs and local variables. Tokens circulating on channels and manipulated by actors are divided into data tokens (carrying actual values, such as pixels for example) and control tokens (acting as structuring delimiters). This approach allows fine-grain processing (down to the pixel level) to be expressed without the need of global control and/or synchronization.

The structure of the actor network is described using a strongly-typed, polymorphic, higher-order, purely functional language called FGN (Functional Graph Notation) [2]. This language allows complex networks to be described in a very abstract concise manner.

The stream-based model of computation allows easy interfacing to a wide range of input-output devices, from cameras delivering synchronous pixel streams to other components in a SoC-based design and exchanging data through shared memory.

**Description.** The proposed toolset is sketched on figure 1. The program, describing both the behavior of each actor and the structure of the actor network is first type-checked and compiled to produce an intermediate representation (IR) in the form of a process network, in which each process is represented as a generalized finite-state machine (GFSM) and channels as unbounded FIFOs. Two back-ends are provided: the first produces SystemC code for simulation and profiling, the second VHDL code for hardware synthesis. Execution of the SystemC code provides informations which are used to tailor the VHDL implementation (for example, the actual size of the FIFOs). A reference interpreter, operating at the abstract syntax level and based on a fully formalized semantics of the language provides results to check correctness of the generated SystemC and VHDL code.

**References**

