International Workshop on Reconfigurable Communication-centric Systems-on-Chip

Darmstadt, Germany, July 10 - 12

D-RECS: A Complete Methodology to Implement Self Dynamic Reconfigurable FPGA-Based Systems

F. Cancare, C. Pilato, A. Cazzaniga, D. Sciuto, M. D. Santambrogio

Politecnico di Milano

Rationale

- **Flexibility**: many emerging products in communication, computing and consumer electronics demand that their functionality remains flexible also after the system has been manufactured.
  - Support of new standards, e.g. in media processing
  - Addition of new features
- **Design support for partial dynamic reconfiguration**: Dynamic self reconfigurable embedded systems are gathering, day after day, an increasing interest from both the scientific and the industrial world. At the same time, however, the need of a comprehensive and easy to use tool which can guide designers through the whole implementation process is becoming stronger.
- **Performance and runtime customization**: reconfigurable computing is intended to fill the gap between hardware and software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware. Therefore it is possible to apply *reconfigurable solutions* to systems such as:
  - biomedical implants i.e., an artificial art control
  - telecommunications i.e., adaptive intelligent routers
  - Moreover: intelligent nanorobot control, artificial audio and vision, intelligent transducers at bio-electronic interfaces,…

Innovative contributions

- Aim of this work is **to provide a fast brain to bit design flow** whose goal is to simplify the dynamic reconfigurable system development process by shifting the designer focus from the architecture point of view to the application point of view. The novelties introduced by the proposed work with respect to the state of the art are summarized hereby:
  - D-RECS, the definition of a complete methodology to implement Self Dynamic Reconfigurable FPGA-based systems. The proposed approach shifts the designer focus from the architecture point of view to the application point of view, providing a fast brain to bit design method- ology;
  - the definition of a reconfiguration unaware model-driven approach for modeling of SDR systems;
  - in the runtime decision of the most suitable implementa- tion (software or reconfigurable hardware) due to runtime conditions;
  - **Increase** the reconfiguration performance via novel techniques, i.e. runtime reconfigurable cores relocation, reconfigurable cores identification, reconfigurable cores reuse

Context definition

- **Reconfigurable Computing**: the ability of altering an architecture (microarchitecture), once it has been deployed, to meet at the best the execution needs
  - **Reconfiguration Controller**: the element that is responsible for the physical implementation of the reconfiguration process i.e., in Xilinx FPGA the ICAP controller
  - **Reconfiguration Manager**: the element that is responsible for the management of a reconfiguration process i.e., in ATMEL the AVR microcontroller, in Xilinx the PPC405
- **Reconfigurable Functional Unit**: functionality that can be plugged and/or unplugged at runtime in an already working architecture
- **Reconfigurable Region**: a portion of the device area used to implement a reconfigurable core
- **Relocation**: the ability of moving a reconfigurable functional unit from a location to a new one

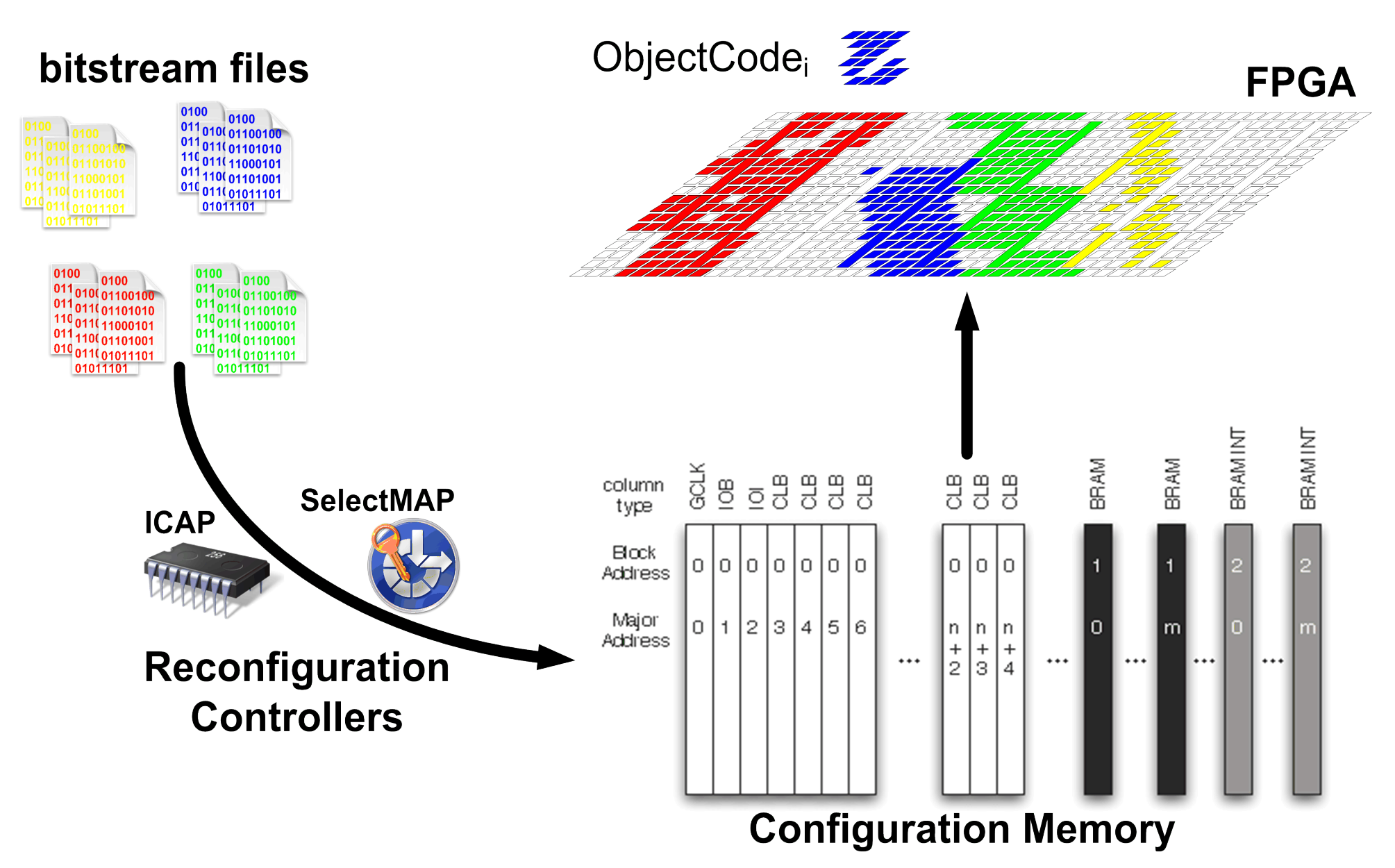


Figure 1 - Xilinx FPGA and configuration memory

Proposed solution

- **Definition** and partial implementation of an entire design flow (Figure 2)
- **Two main contributions** in this thesis can be found in:
  - **Low-level design flow**
    - Generation of the IP-Cores from these cores to be mapped onto the FPGAs
    - Generation of the communication infrastructure
    - Generation of reconfigurable bitstreams
  - **Runtime reconfiguration management**
    - Generation of the runtime reconfiguration manager for internal reconfiguration, if a processor is available
      - These tasks apply to a specific reference architecture constituted by: a static part, reconfigurable slots, communication infrastructure model

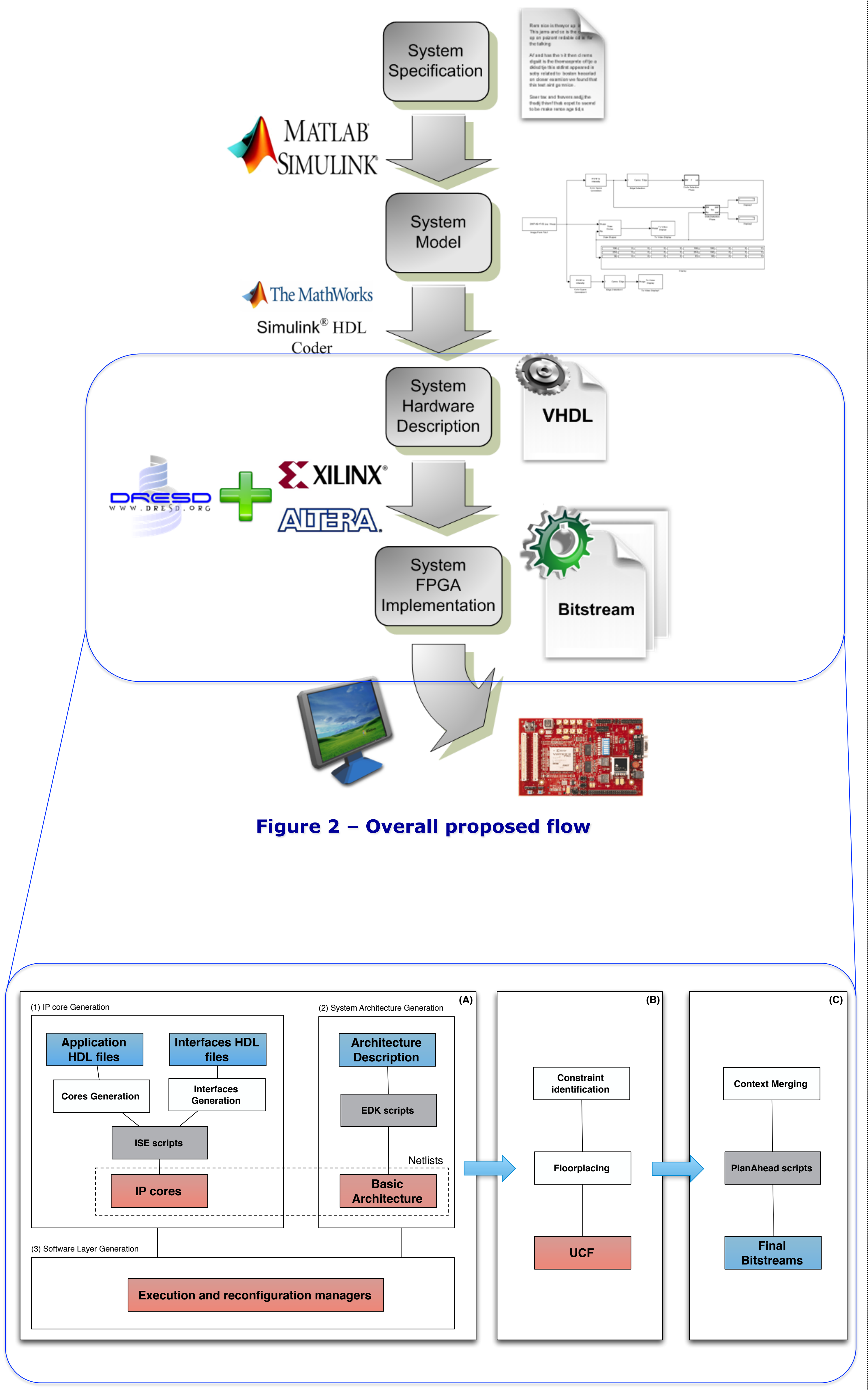
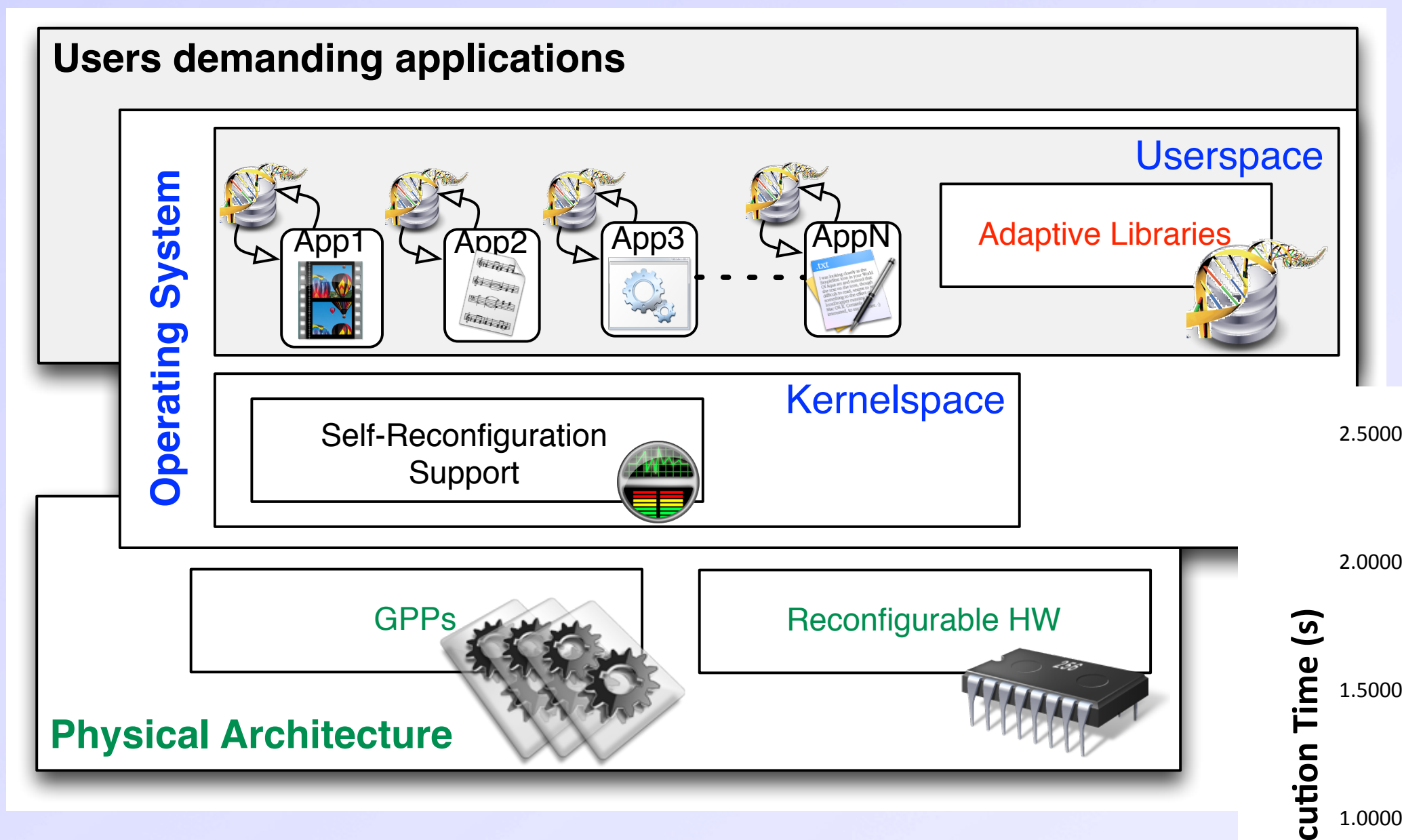
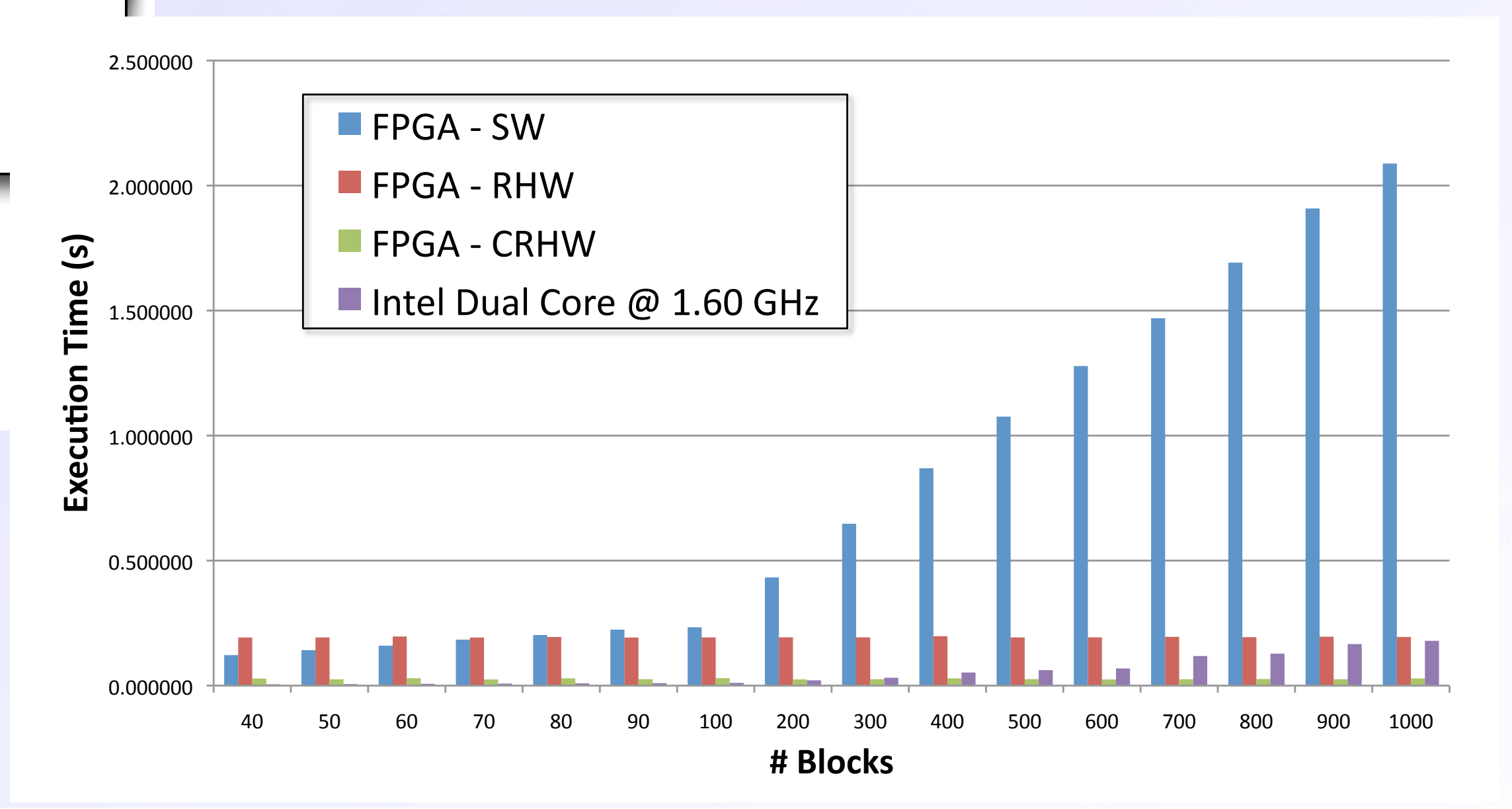


Figure 2 – Overall proposed flow

Runtime reconfiguration management

- **Provide software** support for dynamic partial reconfiguration on Systems-on-Chip running an **operating system**
  - OS customization for specific architectures
  - RFU caching policies to improve the performance
  - Partial reconfiguration process management from the OS
  - Addition and removal of reconfigurable components
  - Automatic loading and unloading of specific drivers for the IP-Cores upon components configuration and/or deconfiguration
  - Hardware-independent interface for software developers based on the GNU/Linux
  - Easier programming interface for specific drivers


Users demanding applications
Operating System
Userspace
App1
App2
App3
AppN
Adaptive Libraries
Kernelspace
Self-Reconfiguration Support
GPPs
Reconfigurable HW
Physical Architecture


Execution Time (s)
FPGA - SW
FPGA - RHW
FPGA - CRHW
Intel Dual Core @ 1.60 GHz
# Blocks