POLITECNICO DI MILANO

Integration of High Level Synthesis Flow in Taste

Marco Lattuada, Fabrizio Ferrandi
Politecnico di Milano

TEC-ED & TEC-SW Final Presentation Days
08 December 2015

Outline

- Introduction
- High Level Synthesis in TASTE
  - Bambu
  - Generated Architecture
- SDL to HDL
- Conclusions

Introduction

+ TASTE provides a framework for the development of safety critical applications
+ FPGAs can be a solution for Space systems because of reconfigurability, computational power and power efficiency

= TASTE included support to integration of HW accelerator

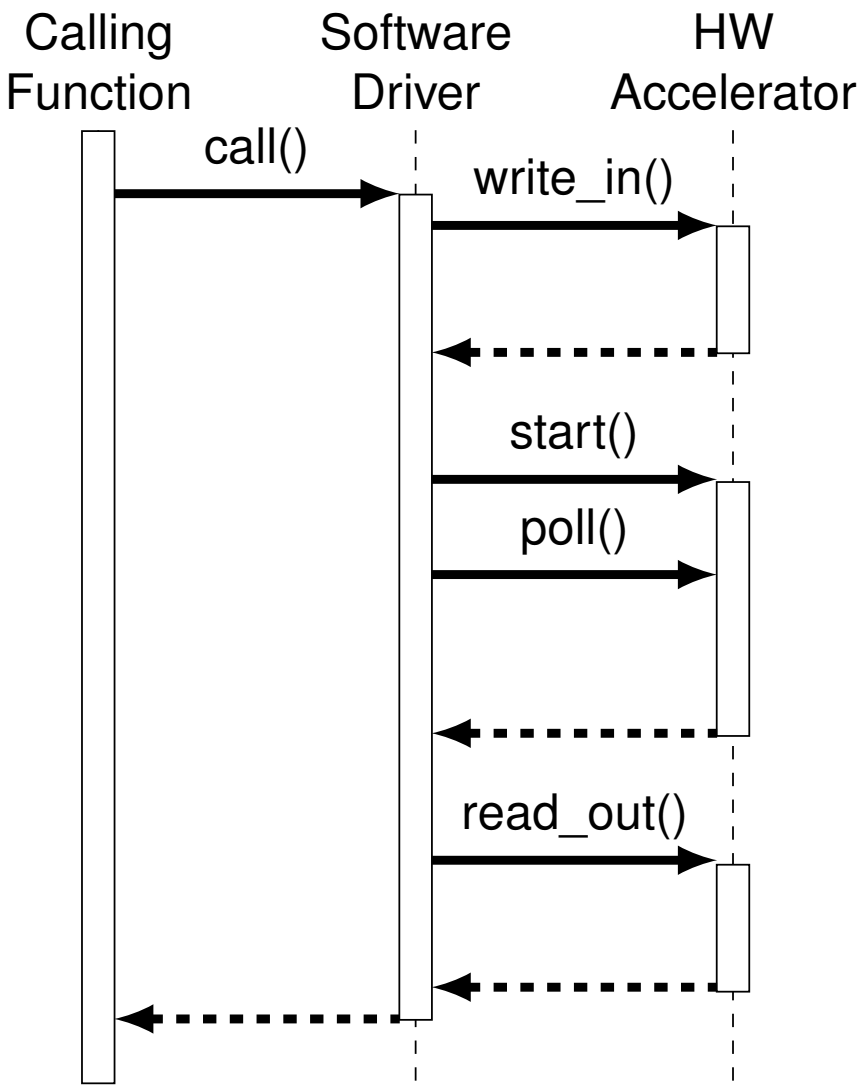
but

DEVELOPER HAS TO DESIGN
IMPLEMENTATIONS OF HARDWARE ACCELERATORS

→ High Level Synthesis generates automatically hardware accelerators starting from high level representation

SOLUTION:
INTEGRATION OF HIGH LEVEL SYNTHESIS IN TASTE

HW/SW Interaction



- Processor connected to FPGA through PCI
- TASTE automatically generates drivers
- Software tasks interface drivers
- Drivers automatically perform copying of input data, polling and copying of output data

High Level Synthesis

- Design flow for the automatic generation of ASIC or FPGA implementation
- Complex design flow, not a simple translation
  - → similar to a compilation flow targeting HDL
- Input: High Level Description - C, C++ or SystemC
- Output: RTL hardware implementations - VHDL or Verilog
- Must be coupled with Logic Synthesis
  - transforms RTL implementation in logic gates implementation
  - performed by FPGA vendor tools

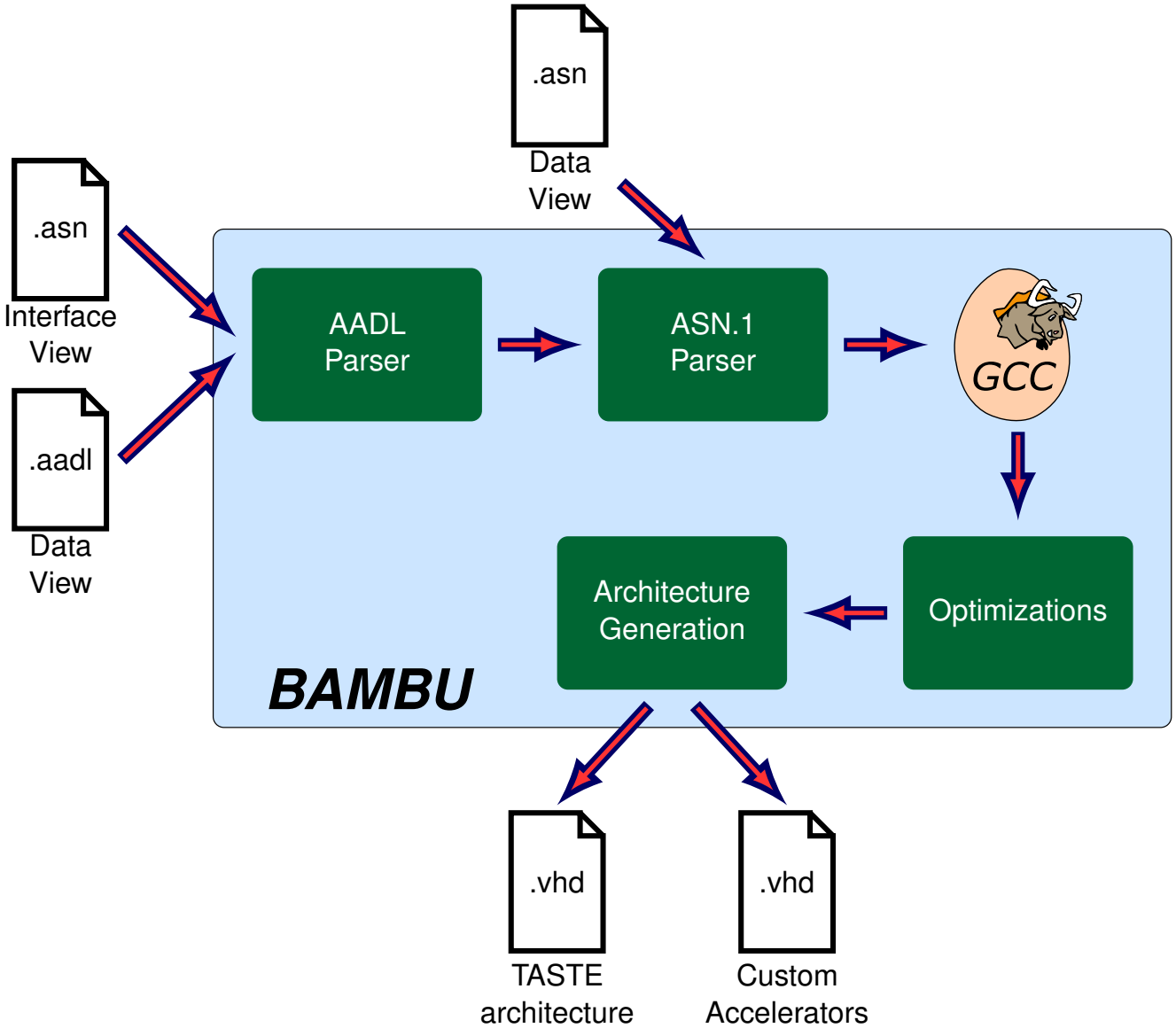
Bambu

- High Level Synthesis tool developed at Politecnico di Milano
- Publicly available under GPL license

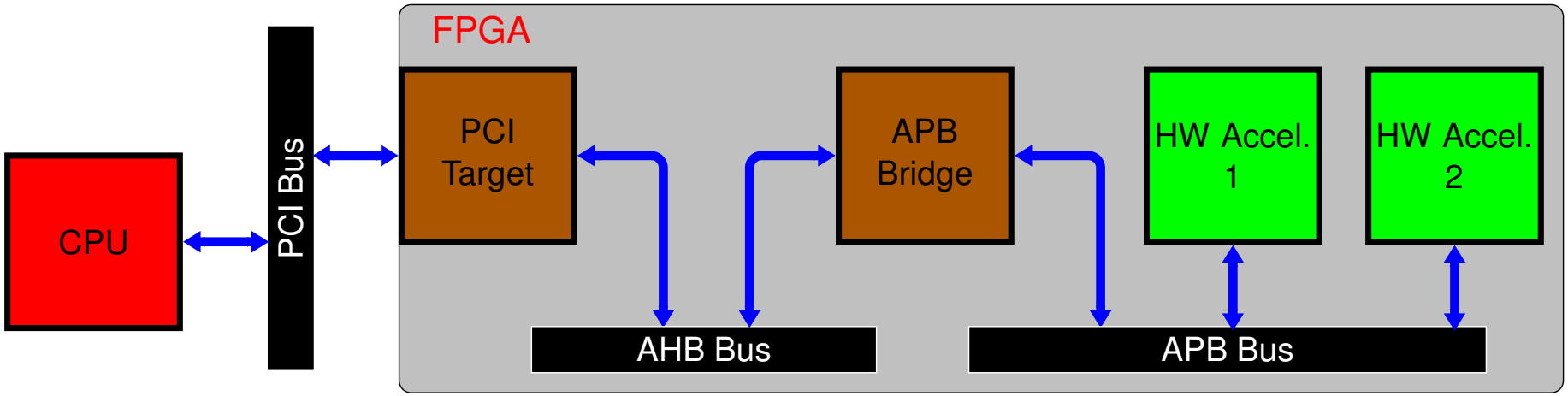
http://panda.dei.polimi.it

- Input is C code:
  - Multiple source files can be used
  - Most of the C constructs are supported (e.g., pointer arithmetic, function pointer, dynamic memory allocation)
  - Existing Hardware Modules can be integrated
- Output is VERILOG/VHDL
- Supports devices from:
  - Altera
  - Lattice
  - Xilinx
- Better/comparable results with respect to commercial tool

Integration of Bambu in TASTE

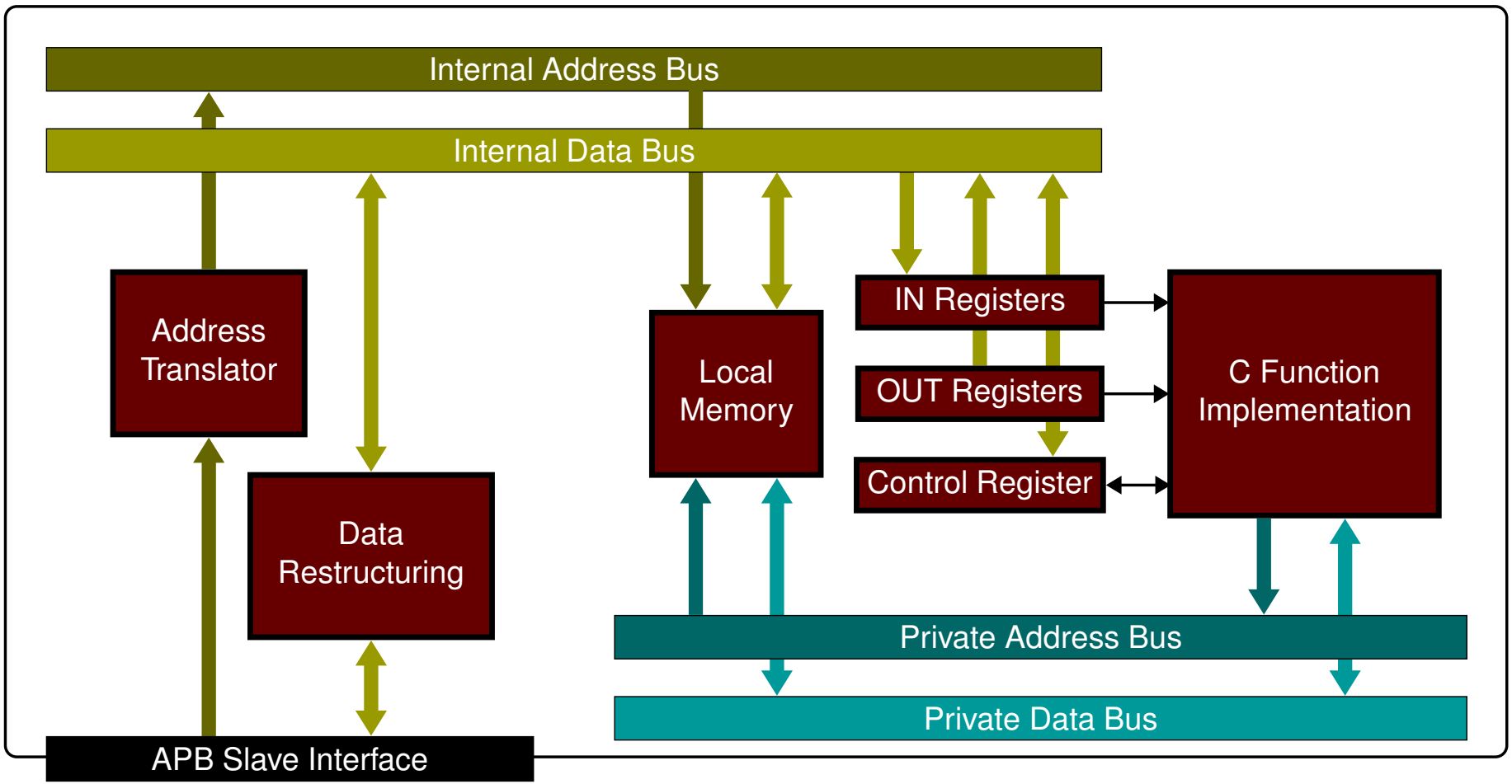


Generated Architecture



- Board is a GR-CPCI-XC4V (Virtex4 XC4VLX FPGA)
- Modules are taken from the GRLIB IP Library
- Two buses: AHB + APB
  - Maintained for backward compatibility

Generated Architecture - Accelerator



Constraints

- TASTE:
  - Safety critical applications coding rules
  - Size of input and output parameters must be fixed
- TASTE FPGA Architecture
  - Overall size of input and output parameters bounded to 256B because of the driver
  - Overall size of input and output parameters bounded to 4KB because of Amba APB
- Bambu:
  - No further constraint

SDL to HDL

Automatic Generation of Hardware Accelerators starting from SDL descriptions performed in two steps:

1. SDL → C
   - performed with OpenGeode
   - A new backend for C has been addedd
   - Supports all the constructs supported by OpenGeode frontend
2. C → VHDL
   - performed with High Level Synthesis

Exploitation of High Level Synthesis

The steps that a designer has to perform to exploit HLS in TASTE starting from C legacy code:

1. *Application Decomposition*
   - → Isolate hardware kernels in separated functions
2. *Algorithm Customization*
   - → Remove all non necessary parameters
3. *Kernels Cleaning*
   - → Remove all the unused code from the kernels
4. *Exchanged Data Identification*
   - → Put all the input/output data in parameters, remove use of global variables

Exploitation of High Level Synthesis

5 *Data View Building*
→ Enrich Data View with the data types used in kernel interfaces

6 *Interface View Building*
→ Building of interface view with the graphical editor

7 *Interface Generation*
→ Automatic Generation of interfaces

8 *C Code Implementation*
→ Interfaces of SW tasks must be filled with actual implementation

9 *Deployment View Building*
→ Building of department view with the graphical editor

10 *System Building*
→ Building of the final system with TASTE

Conclusions and Future Works

- A first prototype of design flow integrating HLS in TASTE has been developed
- The design flow still requires activities by the developer

- Possible future works:
  - Integration of HW/SW partitioning methodologies
  - Tuning of HW/SW interface
  - Development of WCET technique in High Level Synthesis

The presented work has been accepted for publication in the 37th IEEE Aerospace Conference

http://panda.dei.polimi