Choosing IP-XACT IEEE 1685 standard as a unified description for timing and power performance estimations in virtual platforms

Emmanuel Vaumorin (Magillem Design Services)
Motivation

► New needs for global system integration:
  ► Less improvement of design or verification techniques
  ► More management of information and specifications exchange between teams/tools

► Limitations in the current usage:
  ► Expressing system specifications (architecture, timing, power, etc.) with paper or even excel sheet documents
  ► Refinement into the several subparts of hardware platform or software application decoupled from design flows
  ► Complexity of systems is growing

► Proposed solution
  ► Centric representation of the data, using the new IEEE 1685 (IP-XACT)
  ► Backbone for federating the heterogeneous data manipulated by design or verification teams/tools
  ► Requirements traceability for critical systems
  ► In COMPLEX: Centric description of the architecture for federating tools of the framework
    ► MDA software design tools, behavioral synthesis, optimizing cross-compilers, virtual platform generators, automatic design-space exploration tools
Motivation

► Improve management of a development cycle generally focused on improving every step of the process with sophisticated new technologies and using innovative languages or point tools.

► Nevertheless, methodology central services in areas that require complex and expensive development cycle, like SoC production, have identified since years that the most critical issue is mostly related to information management between the several design and verification teams involved in the global process.

► The raising issue with growing complexity of system is less looking for more efficiency in each design or verification process, than providing innovative solution for improving the communication links between these steps.

► Several teams (system architects, hardware and software designers, verification engineers, integrators, etc.) and even companies (system integrator, sub-system developers, component subcontractors) are involved in the process, all working with specific and different concepts, approach, tools, languages, etc.
Inconsistencies of specifications in process cycle

- All teams refer to a written specification and create from this paper documentation the appropriate representation based on their specific activity language and to create the appropriate environment for their tools set.

- As all those steps are performed manually: they are error prone and difficult to maintain in regards with update or modifications.

- The interpretation of these documentations is not always straight as each of the domains use specific conventions which may not be shared by the other actors.

- A lot of possible misunderstanding, undocumented or inconsistent information and consume a lot of time and effort.
Inconsistencies of specifications in process cycle

Solution = provide a centric data representation and corresponding automation mechanisms for handling and managing information through the flow because currently most of those processes implement a document-based transition of the information which leads to multiple issues:

► Each actor of the process has to manually translate documentation to its own specific language (UML, SystemC, HDL, matlab…).
► Modifications which impact the product have to be retranslated manually into a new version of the document which needs to be propagated to upper and lower teams of the cycle.
► Iterations between different levels are difficult due to the fact that modifications have to be manually transmitted to the actors.
► The raising complexity of systems tends increase to the difficulty for the final integrators and project manager to follow changes during the project and thus to ensure a good traceability of requirements and the right implementation of the specifications.
Using a common backbone for information management
Using a common backbone for information management

A standard is available for answering to this issue:
IP-XACT IEEE 1685 is a XML schema allowing a complete structural description of the whole HW+SW system:
- functional, TLM or RTL connections
- Registers and system memory map
- Hierarchical dependencies
- Versions and configuration management
- Links with all file sets repositories

► Commercial solutions exist and provide a complete information backbone to cope with information exchange, synchronization and traceability.

► Share the same information between all the actors, use a common language to describe this information, automate the generation of multiple formats depending on the task needs, automate the update of the golden reference from multiple formats, compute impact of a modification on specifications at any level of the process, perform checks between steps.

► Relies on a computer readable format to exchange all those information and results in a single shared specification which will be enriched at each steps of the process.

► Sharing a common way to describe the exchanged information and use an automated process to extract the piece of information that is relevant for their activity and to automatically translate it into their domain languages or specific tool representations.
1-Import architecture specification from MARTE to IP-XACT
2-Use IP-XACT architecture description to facilitate HW/SW partitioning and application mapping
3-Consistency of architecture description is ensured along the flow
4-Generation of HW virtual platform from IP-XACT
5-Analysis or simulation reports may be linked to several configuration of architecture
6-Design Space Instance parameters may be managed in Magillem SILO linked with IP-XACT
IP-XACT in the COMPLEX’ framework

General Overview

IP-XACT is a XML schema for describing the HW system architecture.

Other schemas or UML profiles can be linked for SW architecture and application mapping.

Also other schemas can be created, dedicated to specific domains (verification, power, timing, PCB, etc.).

IP-XACT is good as centric schema as it handles the structural description of all the system, from functional level down to implementation levels, and managing the hierarchical dependencies.
Thank you very much for your attention.

COMPLEX project partners:

Please visit us at Booth EP16 (COMPLEX) Booth 33 (MAGILLEM)

Project Website: http://complex.offis.de