Asynchronous design methodology using ACT with commercial tools

Filip Hormot (Filip.Hormot@infineon.com),
Marco Scarici (Marco.Scarici@infineon.com),
David Cox (David.Cox@infineon.com)
Introduction

Traditional setup for an analog project

- Analog on top level projects → Digital core included into the analog project
  - Large analog area supported by a small digital core
  - Minimal functionality necessary (Start-up, fault management, NVM, etc.)
Introduction (cont’d)
Inclusion of ACT where to replace a standard digital flow

- Inclusion of ACT with minimal change to the existing processes
- Stability, re-use of existing processes and workflows
Digital design workflow
Concept to netlist methodology for digital design

- Integration of ACT into the design methodology

- Design of and asynchronous circuits → V-model
  - Requirements are set **top-down**
  - Design is done **bottom-up**
    - Iteration of steps for specification violations

---

**Conceptual Design & Architecture**

**Modular division (re-use potential)**

**Design and verification of each module**

**Hierarchical assembly and verification**

**Netlist export and verification in the analog domain**
Synthesis of a CHP process into the analog domain
Overview of the process

https://avlsi.csl.yale.edu/act/doku.php?id=intro_example:cadence_import
Synthesis of a CHP process into the analog domain (cont’d)
Challenges and learning cycles

- Refactoring
  - The generated Verilog netlist is not immediately ready to use
    - Mangle character and prefix removal
    - Removal of error generators (reg/wire definitions)
    - Removal of wrappers → Now available in interact
  - Block reset as a pin → Now available in interact
  - Generated gate I/O naming → Now available in config files

- Import of the control circuitry
  - Config file initial setup → Import all with keeper, IO naming, etc.
  - Gates generated in interact are not the only required gates
    - Additional gates are required to be manually synthesized from the std library
    - Spice netlists do not support component sub-models → Spectre might be better
  - Added effort by additional export procedure (one time activity)
    - No bulk export

- Compliance with our standard gate library
  - Use of cell name mapping → Ongoing
Summary and Next steps

- **Summary**
  - Analog on top level projects
    - Analog/Digital environments are separated
    - Digital circuitry included in the analog domain
  - Inclusion of ACT into existing workflows
    - Re-use of existing structures
    - IP inclusion through Verilog netlists

- **Next steps**
  - Place&route investigations and integration into the methodology
  - Reduction of refactoring effort through technology configuration files
    - Alignment of synthesis rules with design methodologies and (naming) conventions
  - Formalized & automated unit testing
  - Enabling mixed-signal verification domains (using actsim)
In case of inquiries:
- Filip Hormot (Filip.Hormot@infineon.com),
- Marco Scarici (Marco.Scarici@infineon.com),
- David Cox (David.Cox@infineon.com)

Thank you for your attention!