Abstract : testing of an embedded core design

Jan Mennekens **DCIM sprl**
jan@dcim.be www.dcim.be

For the purpose of this talk, an embedded core is an onboard software driven engine. It can take many forms : it can be an IP block of an existing micro-controller (e.g. 8051), or it can be an home-grown Application Specific Instruction Processor (ASIP), or perhaps an extremely flexible type of DMA controller. The important part here is programmability.

More and more, embedded designs incorporating programmable logic feature such an on-board core. There are several reasons for this evolution :

- it buys flexibility : by incorporating an embedded core in the FPGA, there is extra flexibility on multiple levels.
- it costs little : the cost of programmable logic going down, so the price difference between a fixed micro-controller and an embedded one goes down.
- it gets easier to do so : higher-end tools and more flexible design practices make it an attainable goal

However, incorporating an embedded core in the design creates its own set of challenges : the extra flexibility increases the testing complexity. The number of possible states in the hardware explodes, making it nigh-impossible to check them all.
So, in order to keep things manageable, we'll borrow some pages from the software design people. In this talk, we'll try to cover the following topics :

- design for testability from the ground up
- simulation practices : how to keep the simulation model as close as possible to the hardware design without huge effort
- software design practices : unit testing and regression testing, version control and release management

Two existing projects will be used as case study and to explain the techniques used.