Xilinx’s System Generator Toolbox for Matlab

Dr. Vladimir Poučki
DSP Vladimir Poučki S.P., Slovenia
Presentation Outline

• What is System Generator
• Why is it a Great Tool?
• Practical simple example
What is System Generator?

• Toolbox for Matlab/Simulink.
• Designer is Xilinx.
• It is part of almost all Xilinx ISE packages.
• Personal recommendation: install Matlab, and then Xilinx ISE.
• Some fixed-point toolboxes from Matlab are also required.
What is System Generator?

- Just ‘Google’: Xilinx 17966. This provides a clear list of compatible versions of Matlab and Xilinx ISE.
- After installation, this appears as a new Simulink toolbox.
- System Generator => SysGen
- Many functions are available to be used from Matlab command window or scripts.
What is System Generator?

- SysGen toolbox is actually a list of HW blocks, that have almost 1-to-1 matching with HW resources on FPGA, OR the ISE tools can easily synthesize the HW circuits.
Why is it a great tool?

- Many toolboxes designed by HW vendors (like Texas Instruments) have complicated ‘engine’ for code generation based on Simulink model.

- Previous fact results in huge overhead of the generated code => almost unusable for industrial applications.
Why is it a great tool?

• SysGen is NOT TRYING to be smart.

• All building blocks are connected to ISE Core Generator (CoreGen).

• Instance of a single block in SysGen results in a parameterized function call of CoreGen.

• There is NO OVERHEAD.
Why is it a great tool?

• Great ‘intelligence’ is implemented in generating the following:
  - clock wrappers
  - handling of multi cycle paths and multi cycle clocks
  - using sophisticated options of synthesis process: very important.
  - Interconnections of modules => usually a big headache for FPGA designer due to many lines of code that needs to be written
Why is it a great tool?

- Great ‘intelligence’ is implemented in generating the following:
  - generation and usage of ‘ce’ signals is excellent
  - designer does not care if it is Verilog or VHDL
  - can automatically generate test benches (for Modelsim, as an example)
  - easily programmed customized compilation and timing analysis scripts
  - even subtle problems like clock domain crossing is perfectly well solved
Why is it a great tool?

• Great to jump start with implementation of complex processing algorithms.
• Great for inexperienced FPGA programmer since it hides many FPGA specifics.
• Though many have doubts, it can be successfully used even for most complex applications like ‘base station’ signal processing.
• Do not use it for ‘glue logic’ or implementation of HW interfaces (like DDR2 controller).
THANK YOU!

HVALA!