Vivado Results Generation Scripts

Ahmed Ferozpuri

George Mason University
email: aferozpu@gmu.edu

A user of these scripts can choose to implement HDL code of a cryptographic module

- without a wrapper, using the Out-of-Context (OOC) mode of Vivado (OOC mode) [2], or
- with a simple wrapper (aimed at reducing a total number of pins required), using the TopDown mode of Vivado [2].

To generate results for a specific project you must set up the directory structure, list all source files to be included in the project, modify several key files, and finally run a few scripts to generate device specific results. This process is summarized in the step-by-step fashion below. Additionally, Xilinx provides a general tutorial [2], describing the aforementioned design modes in more detail. The instructions below apply to both the OOC mode and the TopDown mode of the results generation, unless otherwise noted.

1. Directory Structure Setup
   (a) Copy the `scripts/VivadoBatch` folder to a new workspace.
   (b) Rename the PROJECT folder to any more specific project name. Note, you can copy and paste this folder as many times as required to accommodate multiple projects.
   (c) Copy all source files to the subfolder "`Sources/hdl`"
2. PRJ File Setup
   For the AEAD_Core implemented in the OOC mode modify `prj/AEAD_Core.prj`. For the CipherCore implemented in the TopDown mode, with a wrapper, modify `prj/CipherCore_Wrapper.prj`, respectively.
   In the respective PRJ file, list names of all source files necessary to implement your circuit (including a possible wrapper). Use the format: vhdl work "`[SRC FOLDER]/[FILENAME]`", e.g., vhdl work "`hdl/AEAD_pkg.vhd`"
3. (OOC Only): Blackbox File
   (a) Create `hdl/AEAD_Core_bb.vhd` with only the entity declaration from `AEAD_Core.vhd`.
4. (Optional): Constraints File
   (a) The target clock frequency or placement constraints for the implementation can be modified in either `AEAD_Core_Wrapper_flpn.xdc` (OOC) or `CipherCore_Wrapper_flpn.xdc` (TopDown). Please note, that for any TopDown implementation, only timing constraints are required, while both timing and placement constraints are used when generating OOC results. For more details, please see "Step 5: Defining the Top-Level Constraints" [2] .

5. Wrapper Files
   (a) Set the appropriate generic values for the wrapper files: `hdl/AEAD_Core_Wrapper.vhd` (OOC) and/or `hdl/CipherCore_Wrapper.vhd` (TopDown)
   (b) Ensure that `hdl/AEAD_Core_Wrapper.vhd` has the correct values of G_W and G_SW, set in accordance with `AEAD_Core.vhd`
6. Script Execution and Result Generation
   (a) Type any of the following four command sequences into Vivado Tcl Shell to generate results in the OOC or TopDown mode, targeting Virtex-7 (v7) or Zynq. Note, you will need to launch a new Vivado Tcl Shell each time if you want to run all four command sequences simultaneously.
   OOC Zynq:

```
vivado -mode batch -source genOOC_zynq.tcl -notrace
vivado -mode batch -source runOOC_zynq.tcl -notrace
```

   OOC v7:

```
vivado -mode batch -source genOOC_v7.tcl -notrace
vivado -mode batch -source runOOC_v7.tcl -notrace
```

   TopDown Zynq:

```
vivado -mode batch -source genTopDown_zynq.tcl -notrace
vivado -mode batch -source runTopDown_zynq.tcl -notrace
```

   TopDown v7:

```
vivado -mode batch -source genTopDown_v7.tcl -notrace
vivado -mode batch -source runTopDown_v7.tcl -notrace
```

   Note, the gen scripts create Synthesis results and/or OOC constraints and the run scripts produce the implementation results, which can be found in the "Implementation" folder for any device specific OOC/TopDown run.
7. Use python script to view results of all implementations
   (a) Navigate to the `python` folder.
   (b) Execute the getResults.py file, which takes the results folder as an input. Usage: `getResults.py [result_folder]`
   (c) All results found in the `result_folder` will be sent to `output.txt` located in the `python` directory.

References

1. Cryptographic Engineering Research Group (CERG) at GMU. (2015, Sep.) GMU Hardware API. [Online]. Available: https://cryptography.gmu.edu/athena/index.php?id=download
2. Xilinx, *Vivado Design Suite User Guide: Hierarchical Design*, April 2015. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/ug905-vivado-hierarchical-design.pdf