EEE6323 – Advanced VLSI Design
Project – Group 2

Title: **Automated Bus Wrapper Integration**

Group Number: Group 2

Group Members: Deepak Harinath Pudukodu Gopalakrishnan, Harihara Vainatheyi Chepuri, Medini Aradhya, Sai Shanmukh Chinimilli

Main Objective:

To develop a framework to automatically augment miscellaneous Ips (Crypto Engines and DSP Blocks) with wishbone compatible wrappers. Also, to demonstrate the automated wrapper generation and corresponding IP functionality through simulation.

Methodology:

Our approach for automating the generation of wishbone compatible IP wrappers involved first parsing the IP block to obtain information on the Input/Output specifications for Crypto and DSP blocks.

In the case of the Crypto cores the python script works with modules in which the plain text is padded with the key and this script can handle different data widths for both input and outputs. Depending on the width of the input and outputs the top wrapper is generated with the wishbone signals .The module is instantiated in the wrapper automatically with the user signals.

In the case of the DSP blocks, the scripts works for all four modules and like the crypto modules it also can handle different input data sizes and different names of the file until it contains the type of dsp core being used. The wishbone skeletal structure is generated and the top filter module is instantiated in this wishbone file with the inputs of the dsp module.

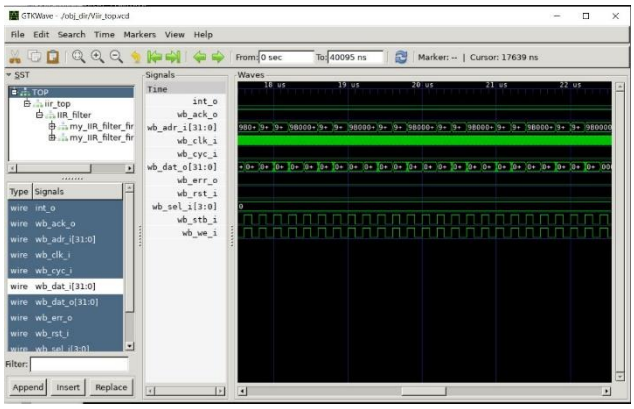
For the wrappers generated by both modules the user defined inputs and outputs have to be connected to the corresponding signals in the wishbone wrapper by the user manually.

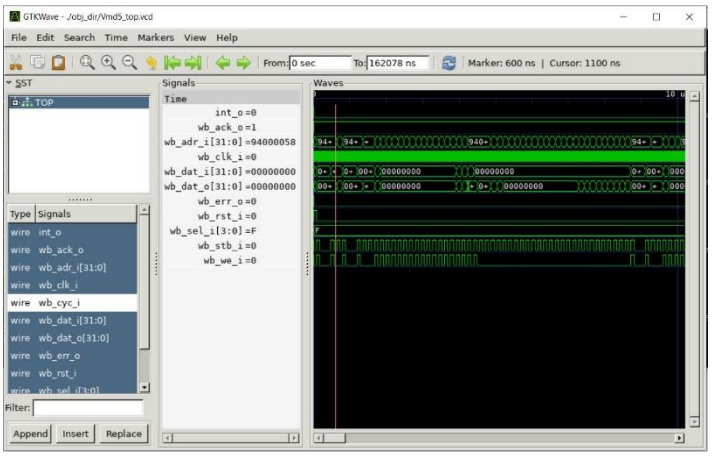
In our implementation we have connected the appropriate signals from the IPs defined by SoCCom tool with the wishbone bus architecture test benches provided and simulated using ‘verilator’ software for their functionality and the output waveforms were observed via ‘GTK Wave’ viewer.

EEE6323 – Advanced VLSI Design
Project – Group 2

Results and Conclusion:

We successfully automated the generation of bus wrapper for both the IPs Ie: Crypto Engine and DSP blocks. The wrappers were then integrated with the wishbone bus architecture. The IP appropriate functionalities were demonstrated through simulations on Verilator and GTK Wave validating the wrapper generated by our python scripts.





Future Scope:

To make the automation more generic, we can remove the chunk of code that is specific to the crypto or the dsp cores and generate the framework of a wishbone wrapper. This wrapper will contain the declarations of the variables of wrapper and the instantiation of the top module of the core with all its inputs and output in it. The user has to insert the wishbone signals he wants to connect to those inputs and outputs along with additional logic as per their requirement.

Work Allocation Statement:

- Deepak Harinath Pudukodu Gopalakrishnan – Worked on DSP core modules and Presentation for Demo.
- Harihara Vainatheyi Chepuri – Worked on generation of automated wrapper for DSP cores.
- Medini Aradhya – Conducted the simulations of both DSP and Crypto cores and Report writing.
- Sai Shanmukh Chinimilli – Worked on generation of automated wrapper for Crypto modules.