Vivado HLS Tool Overview Demo Script

Introduction

This demonstration script provides high-level instructions on the key features to be included with this demo.

Preparation:

- Required files: Necessary files are located at *C:\training\hls\demos\dct*
- Required hardware: None
- Supporting materials: None

Vivado HLS Tool Overview

Action with Description	Point of Emphasis and Key Takeaway

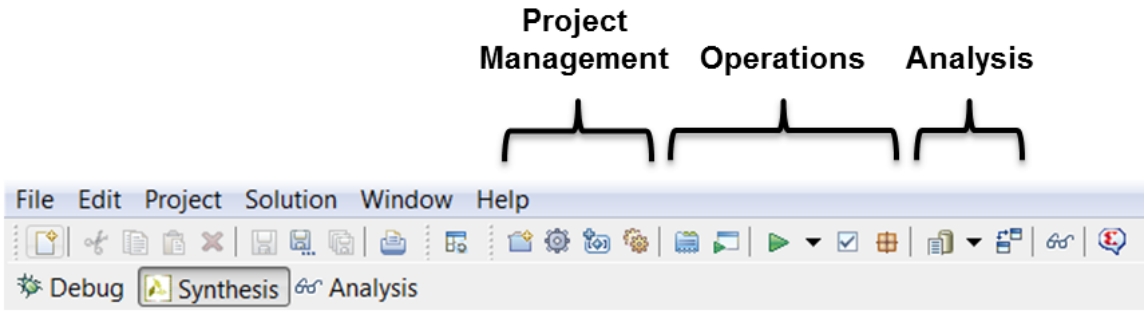
• Launch the Vivado® HLS tool.
• The Vivado HLS Welcome Page has links to:
  • Create a new project
  • Open an existing project
  • Open example projects
  • Open recently opened projects
  • View documentation (tutorials, user guides, and release notes guide)
	Introduce the Welcome page's capabilities:
	• Create New Project: Launch Project Setup Wizard.
	• Open Project: Navigate to an existing project or select from a list of recent projects.
	• Open Example Project: Open Vivado HLS tool example projects (Design Examples & Coding Style Examples) used in the Vivado HLS tool tutorials (*Vivado Design Suite Tutorial: High-Level Synthesis* [UG871])
	• Recent Projects: Single-click access to open recently opened projects.
	• Tutorials: Opens UG871 in DocNav.
	• User Guide: Opens the *Vivado Design Suite User Guide: High-Level Synthesis* (UG902) in DocNav.
	• Release Notes Guide: Opens the *Vivado Design Suite User Guide: Release Notes, Installation, and Licensing* (UG973) for the latest software version.

Action with Description	Point of Emphasis and Key Takeaway
• Launch the New Vivado HLS Project Wizard • This can be done two ways: Click the **New Project** link in the Welcome page (or) select **File** > **New Project** from the menu. • Enter details to create a new Vivado HLS tool Project named *dct_prj* in the *C:\training\hls\demo\dct* directory.	The New Project Wizard guides you through creating an HLS tool project step by step. First the Project Configuration page opens. The Vivado HLS tool is built on the Eclipse software. Hence, in addition to the project name, this file is also the name of the directory in which the project details are stored.
• Specify dct as the top function to be synthesized. • Add "dct.c" as the design file located in the *C:\training\hls\demo\dct* directory.	The Vivado HLS tool will not able to identify the top function to be synthesized and simulated given the complexity of the C-style coding structure. You will need to explicitly specify this. This is not required for SystemC sources, which the Vivado HLS tool will automatically identify. You do not need to specify the location of header files if they are located in the project directory location. If they are not, you will need to specify them explicitly using the CFLAG with -I/project/source/headers option.

Action with Description	Point of Emphasis and Key Takeaway
• Add **dct_test.c** as the testbench file. • Add the testbench stimuli files **in.dat** and **out.golden.dat** as testbench files as well.	The testbench should be a self-checking testbench; i.e., outputs from the top function should compare with the golden results. The testbench should return '0' if the golden results match with DUT results. You can add the folder where your stimuli files are residing by using the Add Folder option.

• Specify solution settings for the default solution:
  • Solution Name: **Solution1**
  • Clock Period: **10ns**
  • Uncertainty: **Leave blank**
  • Part: **Kintex-7 Evaluation Platform** (via board options)
• Click **Finish** to create the new project.
	By default, the New Project Wizard creates a new solution for the project.
	One Vivado HLS tool project can have multiple solutions to allow for design exploration and optimizations.
	The typical solution settings are solution name, clock period, clock uncertainty, and part.
	The clock period used for synthesis is the clock period minus the clock uncertainty. The uncertainty time is to be utilized by FPGA synthesis and the place and route tools. If not specified in ns, the clock uncertainty defaults to 12.5% of the clock period.
	The Vivado HLS tool uses internal models to estimate the delay of the operations for each FPGA device. Hence, a solution targeted with one part may be different from another solution targeted to another part.

Action with Description	Point of Emphasis and Key Takeaway
• Review the Vivado HLS toolbar buttons.	Project Management buttons: • **Create New Project** opens the New Project Wizard. • **Project Settings** allows the current project settings to be modified. • **New Solution** opens the New Solution dialog box. • **Solution Settings** allows the current solution settings to be modified. Operations buttons: • **Index C Source** refreshes the annotations in the C source. • **Run C Simulation** opens the C Simulation dialog box and opens the simulation viewer. • **C Synthesis** starts C source code high-level synthesis. • **Run C/RTL Cosimulation** verifies the RTL output. • **Export RTL** packages the RTL into the desired IP output format. Analysis buttons: • **Open Report** opens the C synthesis report or drops down to open other reports. • **Compare Reports** allows the reports from different solutions to be compared.

Action with Description	Point of Emphasis and Key Takeaway
	
• Run C simulation with the default settings. • Notice the simulation results in the console. • Review the testbench source and verify that the provided testbench is the self-checking testbench.	**Debug:** This compiles the C code and automatically opens the debug perspective. **Build Only:** The C code compiles, but the simulation does not run. **Clean Build:** Remove any existing executable and object files from the project before compiling the code. This option uses a higher level of optimization effort when compiling the design but removes all information required by the debugger.

Action with Description	Point of Emphasis and Key Takeaway
• Run C synthesis. • Review the synthesis report. • Review the synthesis log.	Now that simulation is successful, run C synthesis to generate HDL. Once the synthesis completes, the synthesis report will be opened in the main viewing area for quick review. Notice that the estimated clock period is within the requested clock period. The synthesis report also contains latency and throughput information of the design. The results correspond to the default solution (without any directives). You can further reduce the numbers down by specifying your requirements via directives. The synthesis log will available in the console pane.
• Run C/RTL co-simulation for Verilog HDL. • Accept the default settings in the Co-simulation dialog box. • Review the C/RTL co-simulation results. • Review the co-simulation report.	The C/RTL Co-simulation dialog box allows you to select which type of RTL output to use for verification (Verilog, VHDL or SystemC) and which HDL simulator to use for the simulation (if Verilog or VHDL are selected).
• Export the RTL as IP.	In cases where the Vivado HLS tool uses Xilinx IP in the design, such as with floating-point designs, the RTL directory includes a script to create the IP during RTL synthesis. If the package IP is used, this process is performed automatically by the Xilinx design tools.
• Review the directory structure.	

Summary

In this demo, you learned to create a new project in the Vivado High-Level Synthesis tool. You learned how to create a new project, add source files, specify project settings, run C simulation, and synthesize the design.

References:

- Supporting materials
  - *Vivado Design Suite Tutorial: High-Level Synthesis* (UG871)
  - *Vivado Design Suite User Guide: High-Level Synthesis* (UG902)