

# Vivado IP Flow Demo Script

## Introduction

This demonstration introduces the IP flow, including:

- Upgrading IP
- Customizing IP
- Generating output products
- Instantiating IP
- Identifying the difference between the out-of-context flow (OOC) flow and global synthesis flow

### Preparation:

- Required files: \$TRAINING\_PATH/IP\_Flow/demo/KCU105/verilog
- Required hardware: None
- Supporting materials: None

## Vivado IP Flow

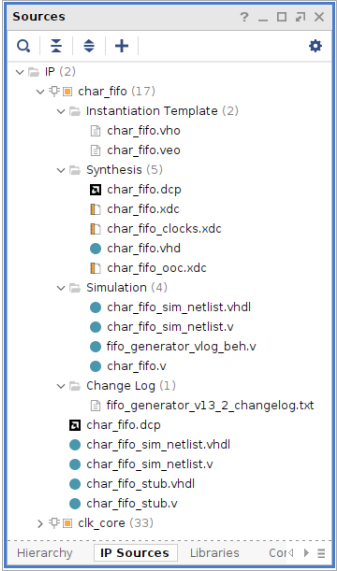
Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"><li>• Launch the Vivado™ Design Suite.</li><li>• Unzip the project using the Tcl Console: <pre>exec unzip \$::env(TRAINING_PATH) / IP_Flow/demo/KCU105/verilog.zip -d \$::env(TRAINING_PATH) / IP_Flow/demo/KCU105/verilog</pre></li></ul>	<ul style="list-style-type: none"><li>• The Getting Started page provides links to:<ul style="list-style-type: none"><li>• Create a new project</li><li>• Open an existing project</li><li>• Open example projects</li><li>• Manage IP</li><li>• Open the hardware manager</li><li>• Visit the Vivado Store</li><li>• View documentation and QuickTake videos for the Vivado Design Suite</li></ul></li></ul>
<ul style="list-style-type: none"><li>• Open the <b>wave_gen.xpr</b> project from the following directory: <pre>\$TRAINING_PATH/IP_Flow/demo/ KCU105/verilog</pre></li></ul>	<ul style="list-style-type: none"><li>• You can easily open an existing Vivado IDE project via the Getting Started page.</li></ul>

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> <li>Go to <b>IP Sources</b> in the Sources window, right-click <b>clk_core</b>, and select <b>Report IP Status</b>.</li> </ul>	<ul style="list-style-type: none"> <li>The IP status tab shows the device part of the IP targeted, under <i>Current Part</i>.</li> <li>In this demo, the <i>clk_core</i> IP is targeted to a Kintex™ 7 series device and the project is targeted to a Kintex UltraScale™ device.</li> <li>The change log for the upgraded IP can also be viewed from IP Status tab.</li> </ul>
<ul style="list-style-type: none"> <li>Right-click <b>clk_core</b> from the IP Sources tab and select <b>Upgrade IP</b>.</li> <li>Click <b>Generate</b> in the dialog box to generate the output products for <i>clk_core</i>.</li> <li>Click <b>OK</b> in the Generate Output Products dialog box.</li> </ul>	<ul style="list-style-type: none"> <li>The IP Status tab after upgrading the IP shows that the <i>clk_core</i> IP is up-to-date.</li> <li>For this demo, you do not need to enable the core container features for this IP.</li> </ul>
<ul style="list-style-type: none"> <li>Rerun the IP Status report from <b>Reports &gt; Report IP Status</b> to notice that the <i>clk_core</i> IP is up-to-date now.</li> </ul>	
<ul style="list-style-type: none"> <li>Select <b>IP Catalog</b> from the Flow Navigator.</li> </ul>	<ul style="list-style-type: none"> <li>The Vivado IP catalog provides consistent, easy access to our IP, including building blocks, wizards, connectivity, DSP, embedded, AXI infrastructure, and video IP from a single common repository regardless of the end application being developed.</li> <li>Note that only one version of IP (i.e., the latest version of IP) will be available in the IP catalog.</li> </ul>
<ul style="list-style-type: none"> <li>Search for and select <b>FIFO Generator</b> from the IP catalog.</li> </ul>	<ul style="list-style-type: none"> <li>The IP catalog contains categories of IP that you can filter and search.</li> <li>You can work with the IP catalog in a variety of ways. You can search using keywords in the search box or browse through the catalog in the various categories.</li> <li>Selecting IP will list all the details of the selected IP: <ul style="list-style-type: none"> <li>Name, version, types of supported interfaces, brief description of IP, IP status, etc.</li> </ul> </li> </ul>

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> <li>Double-click to customize the <b>FIFO Generator</b> IP with the following parameters: <ul style="list-style-type: none"> <li><b>Basic</b> tab &gt; Component Name: <b>char_fifo</b></li> <li>FIFO Implementation: <b>Independent Clocks Built-in FIFO</b></li> <li><b>Native ports</b> tab &gt; Read Mode: <b>First Word Fall Through</b></li> <li>Write width: <b>8</b></li> <li>Write depth: <b>2048</b></li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>The selected IP can be customized by right-clicking and selecting <b>Customize IP</b> or double -click the IP.</li> <li>In the Summary tab of the IP Customization dialog box you can verify that the information is correct and then generate the IP.</li> <li>Once the customization is done, the Generate Output Products dialog box opens.</li> </ul>
<ul style="list-style-type: none"> <li>Click <b>OK</b> and click <b>Generate</b> in the dialog box to generate the output products with the default settings (OOC flow).</li> <li>Click <b>OK</b> if the Generate Output Products dialog box prompts.</li> </ul>	<ul style="list-style-type: none"> <li>By default, the <b>Synthesized Checkpoint (.dcp)</b> is generated, and an Out-of-Context Module Run for the <i>char_fifo</i> IP is added to the Design Runs window.</li> <li>The new run is created and launched to synthesize the IP.</li> <li>When the entire design is synthesized, an HDL stub module is provided in the DCP, which causes a black box to be inferred for the IP.</li> <li>This OOC flow reduces synthesis run times because the IP is synthesized only once. If the top-level design is changed and has to be synthesized, the IP is not re-synthesized.</li> <li>The synthesis log file can be viewed from the properties window by selecting the created IP run from the Design Runs window.</li> </ul>

In the Log table tab next to TCL Console tab, select the **char\_fifo\_synth\_1** and see the log file.



Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> <li>Select the <b>IP Sources</b> tab to examine the output products that are generated.</li> </ul> 	<ul style="list-style-type: none"> <li>The IP Sources window shows the generated output products.</li> <li>By default, the IP will generate: <ul style="list-style-type: none"> <li>Instantiation template</li> <li>Synthesis files <ul style="list-style-type: none"> <li>Constraints files associated with synthesis</li> </ul> </li> <li>Simulation files <ul style="list-style-type: none"> <li>Simulation sources associated with IP</li> </ul> </li> <li>Change log</li> <li>Design checkpoint (.dcp) <ul style="list-style-type: none"> <li>Synthesizes the IP and generates the DCP file in OOC mode</li> </ul> </li> <li>Simulation netlist and stub files</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>View the instantiation template that is generated for <i>char_fifo</i>.</li> <li>Since it is already been instantiated, you can view it in the <b>Hierarchy</b> tab under the top-level file (<i>wave_gen</i>).</li> </ul>	<ul style="list-style-type: none"> <li>After the IP is generated, if the IP is not instantiated in the design it will be added at the same level as the top-level <i>wave_gen</i> module in the Hierarchy tab.</li> <li>The IP has already been instantiated in the design. To view the instantiation, open the <i>wave_gen.v</i> file in the text editor and observe lines 338 to 350.</li> <li>Notice that the Hierarchy, Libraries, and Compile Order tabs are updated to indicate that the IP has been instantiated into the design.</li> </ul>
<ul style="list-style-type: none"> <li>Observe the directory structure of the generated output products for the IP. <ul style="list-style-type: none"> <li>Browse to the <code>\$TRAINING_PATH/IP_Flow/demo/KCU105/verilog/wave_gen.srscs/sources_1/ip</code> directory.</li> <li>Notice that there is one directory for each IP.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>The <i>clk_core</i> directory contains the output products for the <i>clk_core</i> IP.</li> <li>The <i>char_fifo</i> directory contains the output products for the <i>char_fifo</i> IP.</li> </ul>

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> <li>Examine the output products for the <i>char_fifo</i> IP in the IP Sources tab of the Sources view. <ul style="list-style-type: none"> <li>Select the <b>IP Sources</b> tab in the Sources window.</li> <li>Expand the <b>char_fifo</b> and <b>clk_core</b> folders and observe that the outputs products for both IPs are the same.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Within the Vivado IDE Sources view, the two IP appear the same, both listing the output products, all of which can be opened for viewing from within the Vivado IDE.</li> </ul>
<ul style="list-style-type: none"> <li>Generate the output products for <i>char_fifo</i> by using the global synthesis flow. <ul style="list-style-type: none"> <li>Right-click <b>char_fifo</b> in the IP Sources window and select <b>Generate Output Products</b>.</li> <li>Select the <b>Global</b> option in the Synthesis Options section and click <b>Generate</b> in the Generate the output products section.</li> <li>Click <b>OK</b> if the Generate Output Products dialog box appears.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Note that new design run for <i>char_fifo</i> is not created as in the OOC flow.</li> <li>Now, the <i>char_fifo</i> IP output products will be generated while synthesizing the top-level design.</li> </ul>
<ul style="list-style-type: none"> <li>Synthesize the design.</li> </ul>	<ul style="list-style-type: none"> <li>The <i>char_fifo</i> IP gets synthesized along with the top-level design.</li> </ul>
<ul style="list-style-type: none"> <li>Right-click the <b>char_fifo</b> IP and select <b>Remove File from Project</b> to delete the <i>char_fifo</i> IP from the IP Sources window. <ul style="list-style-type: none"> <li>Also, make sure that the IP files are removed from the project directory.</li> </ul> </li> <li>Run a Tcl script to generate the <i>char_fifo</i> IP. <ul style="list-style-type: none"> <li>Run the Tcl script from the Tcl Console by using the following command: <pre>source \$::env(TRAINING_PATH) / IP_Flow/demo/KCU105/verilog/ char_fifo_run.tcl</pre> </li> <li>View the Tcl script and output products generated for <i>char_fifo</i>.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>IP can be created and customized by using Tcl commands such as <code>create_ip</code> and <code>set_property</code>.</li> <li>The <code>generate_target</code> command is used to generate output products.</li> </ul>
<ul style="list-style-type: none"> <li>Select <b>File &gt; Exit</b> to close the Vivado Design Suite.</li> </ul>	

## Summary

This demonstration showed you the IP flow, use of the core container feature, and the difference between out-of-context (OOC) and global synthesis flow.

References:

- Supporting materials
  - *Vivado Design Suite User Guide: Designing with IP* (UG896)
  - Using Core Containers for IP Quick Take video  
<https://www.xilinx.com/video/hardware/using-core-containers-for-ip.html>