

Design Analysis Using Tcl Commands Demo Script

Introduction

In this demonstration, you will perform design analysis by finding objects, object properties, and object connectivity by using Tcl commands.

Preparation:

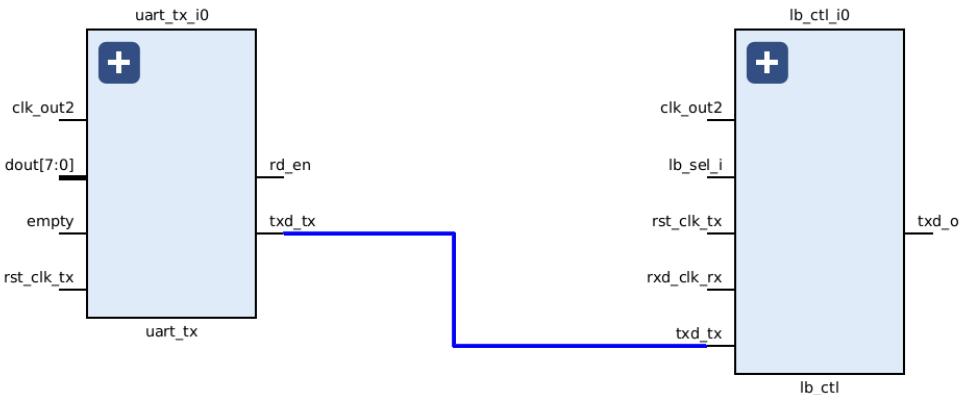
- Necessary project files are located in the following directory:
\$TRAINING_PATH/Tcl_Dsgn_Analysis/demo/KCU105/verilog
- Required hardware: None

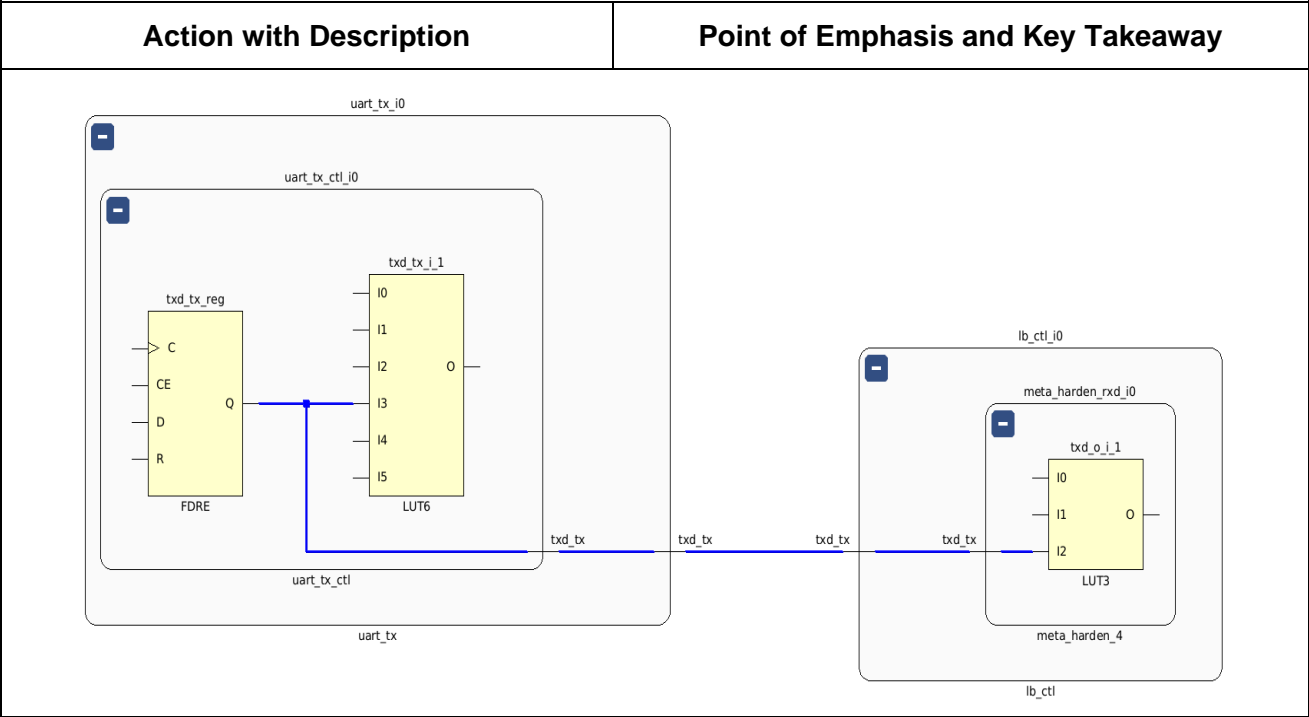
Finding Objects

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Launch the Vivado™ Design Suite. Unzip the project using the Tcl Console: <pre>exec unzip \$:env(TRAINING_PATH)/Tcl_Dsgn_Analysis/demo/KCU105/verilog.zip -d \$:env(TRAINING_PATH)/Tcl_Dsgn_Analysis/demo/KCU105/verilog</pre> Open the wave_gen.xpr project from the following directory: \$TRAINING_PATH/Tcl_Dsgn_Analysis/demo/KCU105/verilog 	<ul style="list-style-type: none"> You can easily open an existing Vivado IDE project via the Getting Started page.
<ul style="list-style-type: none"> Open the implemented design. 	<ul style="list-style-type: none"> You can open the implemented design by using: <ul style="list-style-type: none"> The Flow Navigator The Tcl Console The horizontal toolbar The implemented design shows how the logic will be placed on resources. It allows you to generate various clock and timing report.
<ul style="list-style-type: none"> Is the hierarchy maintained in the implemented netlist? <ul style="list-style-type: none"> Yes, hierarchy is maintained. You can view the netlist hierarchy in the Netlist window (Window > Netlist). You can also review the hierarchy settings for the project in the synthesis settings. 	

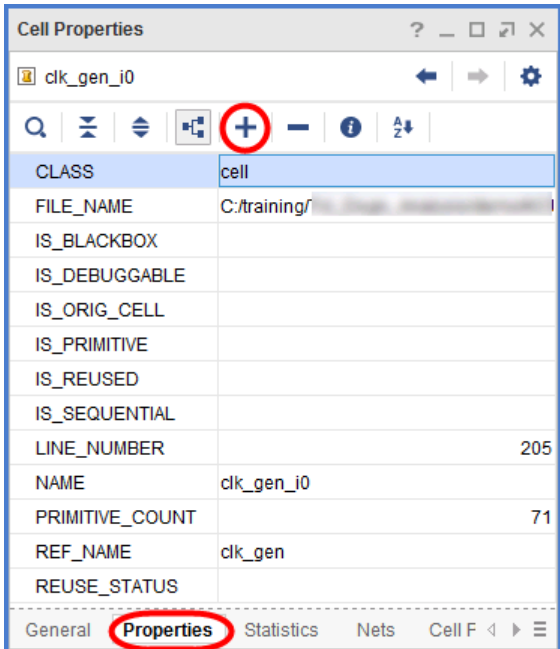
Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>get_cells</code> in the Tcl Console to get all the cells in the design within the current level of the hierarchy. 	<ul style="list-style-type: none"> The Tcl interface provides commands for getting objects. Some of them include: <ul style="list-style-type: none"> <code>get_cells</code>, <code>get_nets</code>, <code>get_ports</code>, <code>get_pins</code> With no options, these commands return a list of all objects at the current level of hierarchy. <ul style="list-style-type: none"> The current level of hierarchy is the top level of the design unless changed by the user. Note: As you enter <code>get_</code> in the Tcl Console, the Vivado Design Suite lists all the commands supported by it.
<ul style="list-style-type: none"> What is the output of the <code>get_cells</code> command? Is this expected? How many cells are returned by the <code>get_cells</code> command? <ul style="list-style-type: none"> The <code>get_cells</code> command returns the list of cells at the current level of hierarchy (in this case <code>top_level</code>). This list does not include any cells from the hierarchical cells (sub-blocks) under the top level. Enter the <code>llength [get_cells]</code> command to determine the number of cells returned by the <code>get_cells</code> command. 	
<ul style="list-style-type: none"> Enter <code>get_cells -hierarchical</code> in the Tcl Console. 	<ul style="list-style-type: none"> This command lists all the cells in the design. The <code>-hierarchical</code> option can be used with cells, pins, or nets.
<ul style="list-style-type: none"> Enter <code>get_cells uart_rx</code> in the Tcl Console. 	<ul style="list-style-type: none"> If no object of that name given in the command exists, the command will return a null list with a warning message. The <code>uart_rx</code> cell is taken as an example here. You can use any other cell.
<ul style="list-style-type: none"> Enter <code>get_cells uart_rx_i0</code> in the Tcl Console. 	<ul style="list-style-type: none"> This returns the cell <code>uart_rx_i0</code> from the current level of the hierarchy.
<ul style="list-style-type: none"> Enter <code>get_cells OBUF_dac_c*_n</code> in the Tcl Console to use a wildcard to match characters. Enter <code>get_cells OBUF_dac_c?_n</code>. 	<ul style="list-style-type: none"> The names given in the command can have wildcards.
<ul style="list-style-type: none"> How did the wildcards <code>*</code> and <code>?</code> match the characters differently in each case? <ul style="list-style-type: none"> The wildcard <code>*</code> matches any number of characters, so the command with <code>*</code> returns two cells <code>OBUF_dac_clr_n</code> and <code>OBUF_dac_cs_n</code>. The wildcard <code>?</code> matches a single character, so the command using <code>?</code> returns <code>OBUF_dac_cs_n</code>. 	

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>get_cells -regexp {[Rr]st.*}</code> in the Tcl console. 	<ul style="list-style-type: none"> Using the <code>-regexp</code> option specifies that the search patterns are written as regular expressions. You can add <code>".*"</code> to the beginning or end of a search string to match any number of characters. This command searches for strings starting with <code>Rst</code> or <code>rst</code> and has any number of characters after that.
<ul style="list-style-type: none"> Enter <code>get_cells -hierarchical uart_baud_gen_*</code> in the Tcl Console. 	<ul style="list-style-type: none"> By default, the <code>get_*</code> commands only search for objects starting at the current level of hierarchy. The option <code>-hierarchical</code> will search at all levels of the hierarchy.
<ul style="list-style-type: none"> Enter <code>get_pins samp_gen_i0/rst_clk_tx</code> in the Tcl Console. 	<ul style="list-style-type: none"> Pin names are based on the instance they belong to. When searching for pins, you must use the hierarchy separator to separate the instance name and the pin name patterns. The command returns the <code>clk_tx</code> pin of the <code>samp_gen_i0</code> instance.
<ul style="list-style-type: none"> Enter <code>get_pins -hierarchical */C</code> in the Tcl Console. 	<ul style="list-style-type: none"> This command returns the C pins of any cell in the design as the wild card <code>*</code> is used to match pin names. Note that the pin names look hierarchical, but they are not. <ul style="list-style-type: none"> For example, pin C of the instance. <code>signal_meta_reg</code> is named as <code>signal_meta_reg/C</code>. This does not mean that the C pin is inside the hierarchy of the instance <code>signal_meta_reg</code>. The final <code>/</code> in a pin name is not a hierarchy separator, it is part of the pin name.

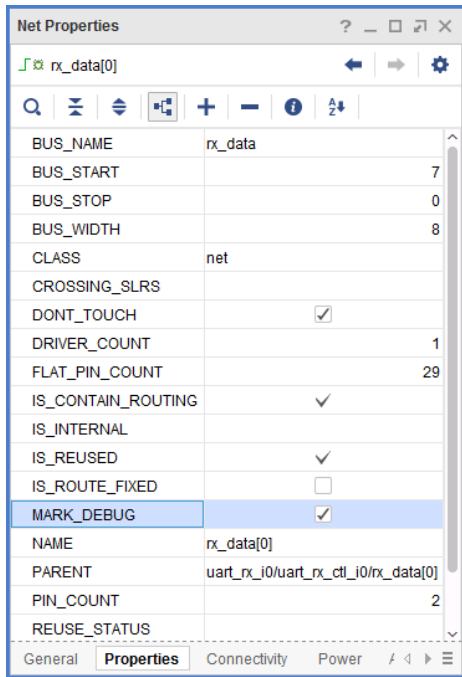
Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>get_nets txd_tx</code> in the Tcl Console. Enter <code>show_schematic [get_nets txd_tx]</code> in the Tcl Console. Enter <code>select_objects [get_nets txd_tx]</code> in the Tcl Console. 	<ul style="list-style-type: none"> The <code>txd_tx</code> net is used as an example here, you can choose any other net as well. The <code>get_nets txd_tx</code> command returns the net <code>txd_tx</code>, if present in the current level of hierarchy. The <code>show_schematic</code> command creates a Schematic view containing the specified design objects. The <code>select_object</code> command highlights the selected object in the GUI (including in the Schematic view).
	
<ul style="list-style-type: none"> Enter <code>get_nets txd_tx -segments</code> to return all the segments of the net <code>rx_data[0]</code>. Enter <code>show_schematic [get_nets txd_tx -segments]</code> in the Tcl Console. Enter <code>select_objects [get_nets txd_tx -segments]</code> in the Tcl Console. 	<ul style="list-style-type: none"> A single logical net can have multiple segments. <ul style="list-style-type: none"> Within each hierarchical object, the net can have a different segment. Each segment is a different net object. The <code>get_nets</code> command with <code>-segments</code> option can be used to find the segments of a net. One of the segments will be considered the parent segment. All the other segments will have the PARENT property set to refer to this segment. Accessing the properties will be covered in the next part of the demo.

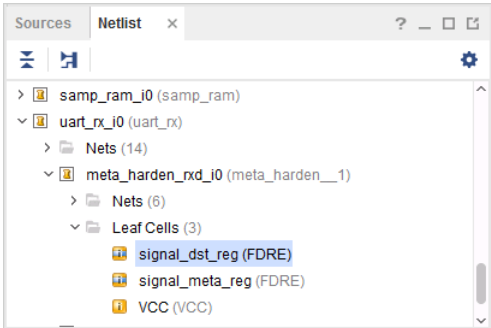
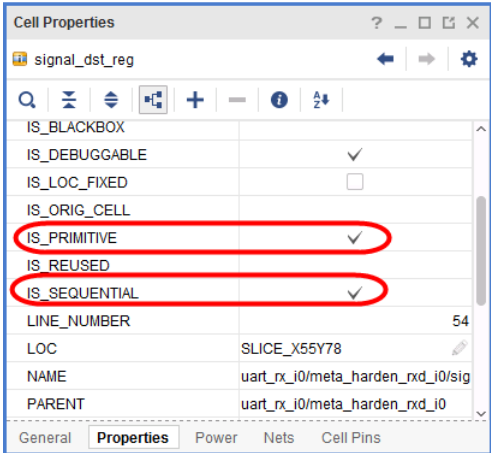


Object Properties

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Select any cell from the netlist (clk_gen_i0, for example). View the class type and other properties from the Cell Properties window. 	<ul style="list-style-type: none"> Object properties can be seen by selecting an object. <ul style="list-style-type: none"> The selection can be done in a schematic or any object list. Only the most common properties and properties that are at their non-default value are shown in the Cell Properties window. Additional defined properties can be added to the view by right-clicking in the window and selecting Add Properties or by clicking the + sign. A popup window will show all possible properties that can be added to the view.
<ul style="list-style-type: none"> Enter <code>report_property [get_cells clk_gen_i0]</code> in the Tcl Console. Enter <code>report_property -all [get_cells clk_gen_i0]</code> in the Tcl Console. 	<ul style="list-style-type: none"> The <code>report_property</code> command returns the property name, property type, and property value for all of the properties on a specified object or class of objects. You can specify objects for <code>report_property</code> by using the <code>get_*</code> series of commands to get a specific object. The <code>-all</code> option returns all of the properties for an object even if the property value is not currently defined.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>list_property -class net</code> in the Tcl Console to list all the properties of <code>net</code>. 	<ul style="list-style-type: none"> This command will list any property that can be had by any object of class-type net. All objects of a given class have the same set of properties. You can use <code>report_property</code> instead of <code>list_property</code> to get the description and type of the property.
<ul style="list-style-type: none"> Enter the following commands in the Tcl Console and observe the output: <ul style="list-style-type: none"> <code>get_property bus_width [get_nets rx_data[0]]</code> <ul style="list-style-type: none"> This command returns the value of the property <code>bus_width</code> of the object <code>net rx_data[0]</code>. <code>get_property slew [get_nets rx_data[0]]</code> <ul style="list-style-type: none"> This command returns nothing. Why? 	<ul style="list-style-type: none"> All property values can be obtained via the <code>get_property</code> Tcl command. <code>get_property</code> gets the current value of the named property from the specified object or objects. If the property is not currently assigned to the object, or is assigned without a value, then the <code>get_property</code> command returns nothing, or the null string.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter the following command in the Tcl Console: <ul style="list-style-type: none"> <code>set_property MARK_DEBUG 1 [get_nets {rx_data[0]}]</code> <ul style="list-style-type: none"> This command sets the property <code>mark_debug</code> of the net <code>rx_data[0]</code>. How can you verify whether the property is set or not? <ul style="list-style-type: none"> Use the <code>get_property</code> command: <pre>get_property MARK_DEBUG [get_nets {rx_data[0]}]</pre> 	<ul style="list-style-type: none"> Properties that are not read only can be set via the <code>set_property</code> Tcl command. The <code>set_property</code> command sets the property on object. Properties can also be modified directly in the Attributes pane of the Instance Properties window in the GUI by right-clicking the particular net and selecting Net Properties.  <ul style="list-style-type: none"> Note that the property you set through the Tcl command is reflected in the GUI. This shows the interoperability between the Tcl interface and the GUI.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Select any primitive cell (uart_rx_i0/meta_harden_rxd_i0/Leaf Cells/signal_dst_reg, for example) from the Netlist window and view the properties from the Cell Properties window.  	<ul style="list-style-type: none"> All objects have a CLASS property, which determines which type of object it is. NAME is the full hierarchical name of the object. IS_PRIMITIVE indicates that it is a primitive cell (as opposed to a hierarchical cell). IS_SEQUENTIAL indicates that it is a clocked element. REF_NAME is the name of the library cell, module, or entity of which this object is an instance. The library cell attributes (i.e., MMCM, Block RAM, ...) are all properties. Similarly, you can select any hierarchical cell. A hierarchical cell is a module or block that contains one or more additional levels of logic and eventually concludes at primitive cells.
<ul style="list-style-type: none"> Enter <code>report_property [get_ports led_pins[0]]</code> in the Tcl Console. <p>Note: You can open the elaborated design and select the port from the schematic and view the properties from the Property window</p>	<ul style="list-style-type: none"> DIRECTION is the direction of the port. I/O attributes are all properties of the port: <ul style="list-style-type: none"> IOSTANDARD, SLEW, DRIVE, PULLUP, PULLDOWN, KEEPER PACKAGE_PIN is the FPGA pin used for the port.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Select any pin from the schematic and view its properties or use the <code>report_property</code> and <code>get_pins</code> commands to view the properties of the pin. <ul style="list-style-type: none"> <code>report_property [get_pins rst_gen_i0/rst_clk_rx]</code> This command returns the properties of the pin <code>rst_gen_i0/rst_clk_rx</code>. 	<ul style="list-style-type: none"> DIRECTION indicates if this pin is an IN, OUT, or INOUT. IS_LEAF indicates if this is the pin of a leaf cell or a pin of a hierarchical cell. IS_CLOCK indicates if this is a clock pin to a clocked object. IS_RESET, IS_PRESET indicate if this pin is a reset or preset pin of a clocked object. IS_ENABLE indicates if this pin is a clock enable pin of a clocked object.
<ul style="list-style-type: none"> Enter <code>create_property -type bool MY_PROPERTY cell</code> in the Tcl Console. Creates the property <code>MY_PROPERTY</code> for all objects of the class <code>cell</code>, which can accept a Boolean value. 	<ul style="list-style-type: none"> New properties can be added to a class of objects with the <code>create_property</code> Tcl command. <code>create_property -type <type> <name> <class></code> <ul style="list-style-type: none"> <code><type></code> can be <code>string</code>, <code>int</code>, <code>long</code>, <code>double</code>, or <code>bool</code>. <code><class></code> can be <code>cell</code>, <code>pin</code>, <code>net</code>, or <code>port</code>. <code><name></code> is the name of the new property to be created. Once added to a class, the new property can be used in the same way as other properties.
<ul style="list-style-type: none"> Enter <code>filter [get_ports] {DIRECTION == IN}</code> in the Tcl Console. 	<ul style="list-style-type: none"> Properties can be used to filter lists of objects with the <code>filter</code> Tcl command. <code>filter <objects> <filter_expression></code> <ul style="list-style-type: none"> <code><objects></code> is a list of objects to filter. <code><filter_expression></code> is the filter expression to use. Returns the list of objects from the <code><objects></code> list that match the <code><filter_expression></code>.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>get_cells -hierarchical -filter { PRIMITIVE_TYPE =~ *MMCME3_ADV* }</code> in the Tcl Console. This command returns cells with the <code>primitive_type</code> property matching <code>*MMCME3_ADV*</code>. 	<ul style="list-style-type: none"> You can also filter a list of objects based on the property with the <code>-filter</code> option of the <code>get_*</code> command. <code>get_cells -filter <filter_expression></code> <ul style="list-style-type: none"> All options of the <code>get_cells</code> command can be used (<code>-hierarchy, ...</code>) Filter expressions are strings. They are usually enclosed in <code>{ }</code>, but <code>" "</code> can be used as well.
<ul style="list-style-type: none"> Select Edit > Find. Select the cells in the Find section. Select PRIMITIVE_TYPE is CLOCK.PLL.MMCME3_ADV in the Properties section. Click OK to find the objects matching the search pattern. 	<ul style="list-style-type: none"> You can use the Edit > Find command or the <Ctrl + F> keyboard shortcut to search for design objects. The command bar in the Find window shows the Tcl command that is run to populate the Find Results window.

Object Connectivity

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Select wave_gen > Nets > clk_pin_p from the Netlist window. Press <F4> to show the schematic of the clk_pin_p net. 	<ul style="list-style-type: none"> Note the pins and cells connected to the net. How can you get pins connected to a particular net?
<ul style="list-style-type: none"> Enter <code>get_pins -of_objects [get_nets clk_pin_p]</code> in the Tcl Console. <ul style="list-style-type: none"> This command returns the pins connected to the net clk_pin_p. Note that this net is connected to a port at one end and a pin at another end. Hence it results in only one pin name; i.e., clk_gen_i0/clk_pin_p. 	<ul style="list-style-type: none"> Objects within the database are related to each other to form the connectivity of the netlist. This connectivity can be accessed by using the Tcl <code>get_*</code> commands. The <code>-of_objects</code> option returns the list of objects connected to the specified object. The <code>-of</code> option is the same as <code>-of_objects</code>. This enables you to find objects based on netlist connectivity.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>get_pins -of_objects [get_nets clk_pin_p] -leaf</code> in the Tcl Console. How was the output different from the previous command? 	<ul style="list-style-type: none"> When finding the pins connected to a net that crosses a hierarchical boundary, you can find either the pins of the hierarchical objects or the pins of the primitive objects. This is controlled by the use of the <code>-leaf</code> option. Without <code>-leaf</code>, the hierarchical objects pins will be returned. With <code>-leaf</code>, the pins of the primitive objects will be returned, regardless of where they are in the hierarchy. <div data-bbox="841 743 1409 1289" data-label="Diagram"> <p>The diagram shows a hierarchical structure. A large orange rectangle represents a top-level object. Inside, there are two smaller rectangles. The left one contains a cell 'q' with an input pin 'i2'. The right one contains a cell 'a' with an input pin 'i1'. Between them is a cell 'c'. A green line labeled 'x2' connects 'q' to 'c', and another green line labeled 'x1' connects 'c' to 'a'. Dashed red arrows point from the command 'get_pins -of [get_nets x1] -leaf' to the net 'x1' and from 'get_pins -of [get_nets x1]' to the net 'x2'.</p> </div> <ul style="list-style-type: none"> You can expand the cells from the schematic of <code>clk_pin_p</code> to understand the output.
<ul style="list-style-type: none"> Select wave_gen > Nets > char_fifo_din from the Netlist window and press <F4> to view the schematic. 	<ul style="list-style-type: none"> What would be the Tcl command to get cell <code>resp_gen_i0</code> from the cell <code>char_fifo_i0</code> based on object connectivity?

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Multiple <code>get_*</code> commands can be nested with the <code>-of_objects</code> option to perform arbitrarily complex searches through the design database. 	<ul style="list-style-type: none"> Apart from <code>resp_gen_i0</code>, what else did the command return? And why? <ul style="list-style-type: none"> The <code>get_nets</code> in the command would return more than one object, and hence the resulting list of objects would include far more than just <code>resp_gen_i0</code>.
<ul style="list-style-type: none"> Select wave_gen > lb_ctl_i0 > leaf_cells > txd_o_reg from the Netlist window and view the schematic. 	<ul style="list-style-type: none"> Using the schematic, what are the fan_in objects of pin D of the <code>txd_o_reg</code> cell? <ul style="list-style-type: none"> Double-click the cell pin D to trace the source. Double-click the three inputs of the LUT3.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Enter <code>all_fanin [get_pins lb_ctl_i0/txd_o_reg/D]</code> in the Tcl Console. If you want to visualize the output of the above command on multiple lines, you can use the built-in command <code>join</code> to insert a "newline" character in between each element of the list returned by the command <ul style="list-style-type: none"> <code>join [all_fanin [get_pins lb_ctl_i0/txd_o_reg/D]] \n</code> 	<ul style="list-style-type: none"> The <code>all_fanin</code> command returns a list of pins that can combinatorially reach the specified pin, port, or net.
<ul style="list-style-type: none"> Use the <code>select_objects</code> command to select the output of the above command in the GUI: <ul style="list-style-type: none"> <code>select_objects [all_fanin [get_pins lb_ctl_i0/txd_o_reg/D]]</code> 	<ul style="list-style-type: none"> Did the command select all the pins you see in the schematic? <ul style="list-style-type: none"> You can observe that the <code>uart_tx_i0/uart_tx_ctl_i0/txd_tx_reg/Q</code> pin is not highlighted as this pin is not in the same level of hierarchy as <code>lb_ctl_i0/txd_o_reg/D</code>.
<ul style="list-style-type: none"> Enter <code>select_objects [all_fanin [get_pins lb_ctl_i0/txd_o_reg/D] -flat -startpoints_only]</code> in the Tcl Console. 	<ul style="list-style-type: none"> When the <code>-startpoints_only</code> option is used, none of the intermediate points in the fan-in network are returned. This option can be used to identify the primary driver(s) of the sinks. Similarly, the <code>-only_cells</code> option returns only the cell objects that are in the fan-in path of the specified sinks. This does not return pins or ports.
<ul style="list-style-type: none"> Enter <code>all_fanout [get_pins lb_ctl_i0/txd_o]</code> in the Tcl console. 	<ul style="list-style-type: none"> The <code>all_fanout</code> command returns a list of all objects that can be combinatorially reached from a specified pin, port, or net. Has similar options as <code>all_fanin</code>. Use <code>-endpoints_only</code> to restrict it to sequential cells. Use different options as in <code>all_fanin</code> and observe the results.
<ul style="list-style-type: none"> Close the implemented design and exit the Vivado Design Suite. 	

Summary

This demonstration illustrated how to find objects, and how to observe objects properties and object connectivity by using Tcl commands.

References:

- Supporting materials
 - *Vivado Design Suite Properties Reference Guide* (UG912)
 - *Vivado Design Suite User Guide: Using Tcl Scripting* (UG894)
 - *Vivado Design Suite Tcl Command Reference Guide* (UG835)