Introduction to Zynq Hardware

Lab 4

Using TCL in Vivado Embedded Designs

November 2013
Version 02

Lab 4 Overview

This lab will give users a brief introduction to various TCL commands that can be run to modify the project as well as archive the project. It's wise to understand TCL as all Xilinx tools run TCL under the hood. This can be powerful for scripting commands for design flows that require several steps. Additionally, we'll explore some of the files that have been created so far.

Lab 4 Objectives

When you have completed Lab 4, you will know how to do the following:

- Open and close block designs using TCL
- Execute simple TCL commands to manipulate IP Integrator block designs
- Export block design to a TCL file on disk

Experiment 1: Explore Project files

This experiment explores some of the files created thus far.

Experiment 1 General Instruction:

Open Windows Explorer and browse the files created by Vivado to this point.

Experiment 1 Step-by-Step Instructions:

1. <Optional> If you did not complete Lab 3 or wish to start with a clean copy, delete the `ZynqDesign, and SDK_Workspace` folders in the `ZynqHW/2013_3` folder. Then unzip **`Solutions\ZynqHW_Lab3_Solution.zip`** to the `2013_3` folder. If you have 7-zip installed, you can do this by right-clicking and dragging **`ZynqHW_Lab3_Solution.zip`** to the `2013_3` folder. Select **7-Zip → Extract Here**.

2. Launch **Windows Explorer** (Windows Key + E).

3. **Navigate** to the Speedway Project folder:

C:\Speedway\ZynqHW\2013_3\ZynqDesign

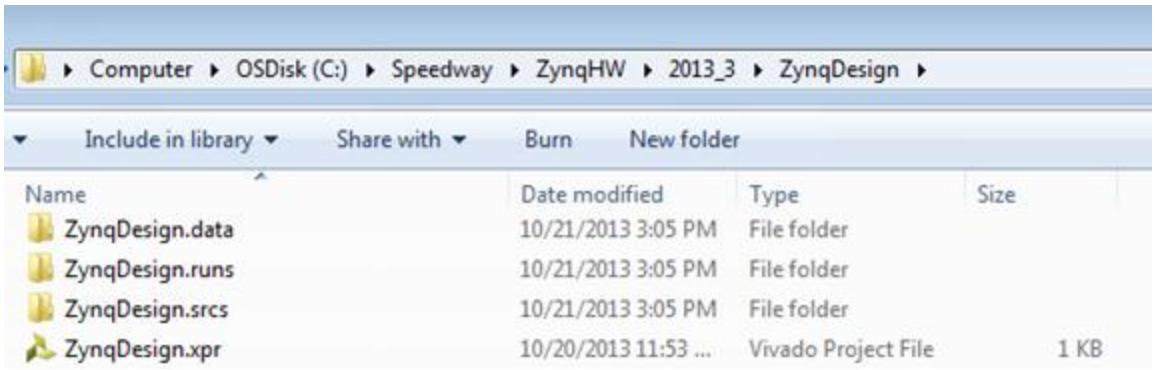


Figure 1 - Project Directory Contents

4. There is only one source file here, **ZynqDesign.xpr**. This is our Vivado project.

5. Our current design is very basic thus far, so there are not many interesting files here. Though the files we have created exist in .SRCS folder. **Navigate** to the following folder:

ZynqDesign.srcs\sources_1\bd\Z_system

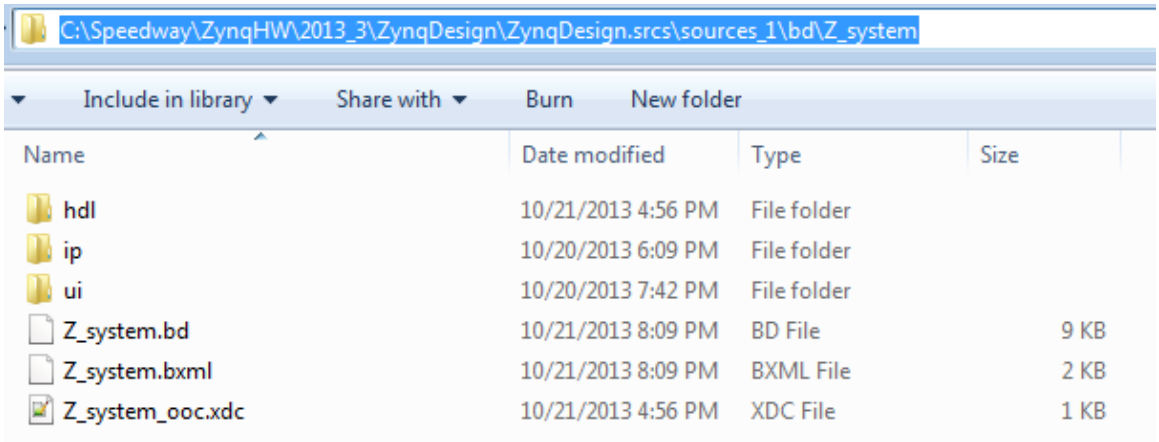


Figure 2 - Project Sources

The only project source we have so far is our block diagram and it can be found in this directory.

6. **Browse** into the HDL folder, you will see our top-level system wrapper as well as our block diagram wrapper.

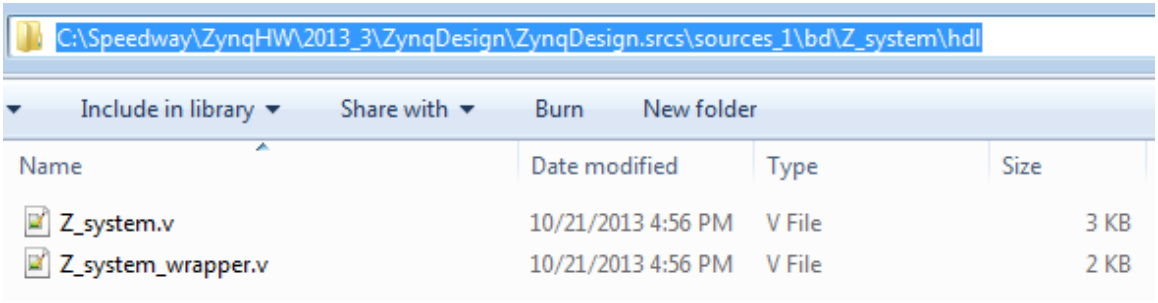


Figure 3 - HDL Wrapper Files

As we progress through the labs, keep checking back into these folders to see what files are created.

Experiment 2: TCL Overview

This experiment will illustrate how to use TCL commands to navigate through Vivado and manipulate our project.

Experiment 2 General Instruction:

Open a Vivado project using TCL commands as well as open block diagram, add remove ports, query design and export block design to a TCL file.

Experiment 2 Step-by-Step Instructions:

1. Launch **Vivado**.

2. Take note from the previous Experiment 1 where the Vivado project is located on the PC. _____

3. Notice the TCL console is available immediately and commands can be entered without opening a project. **Change the working directory** to the Vivado Project directory by entering the following command: (Note: slashes for the directories are forward slashes, not backslashes as Windows uses.)

   cd C:/Speedway/ZynqHW/2013_3/ZynqDesign

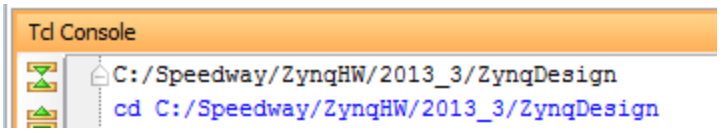


Figure 4 - Change Working Directory

4. Type **pwd** to print the current working directory

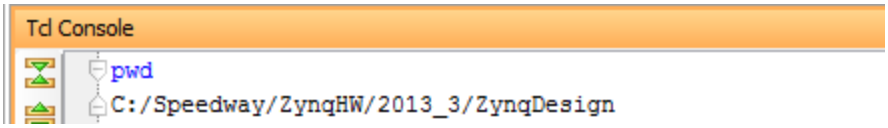


Figure 5 - Print Working Directory

5. In the TCL console open the Vivado project by typing "**open_p**". As TCL commands are entered possible options appear. When open_project appears hit tab to auto-complete the string. Then enter our Vivado project name.

open_project ZynqDesign.xpr

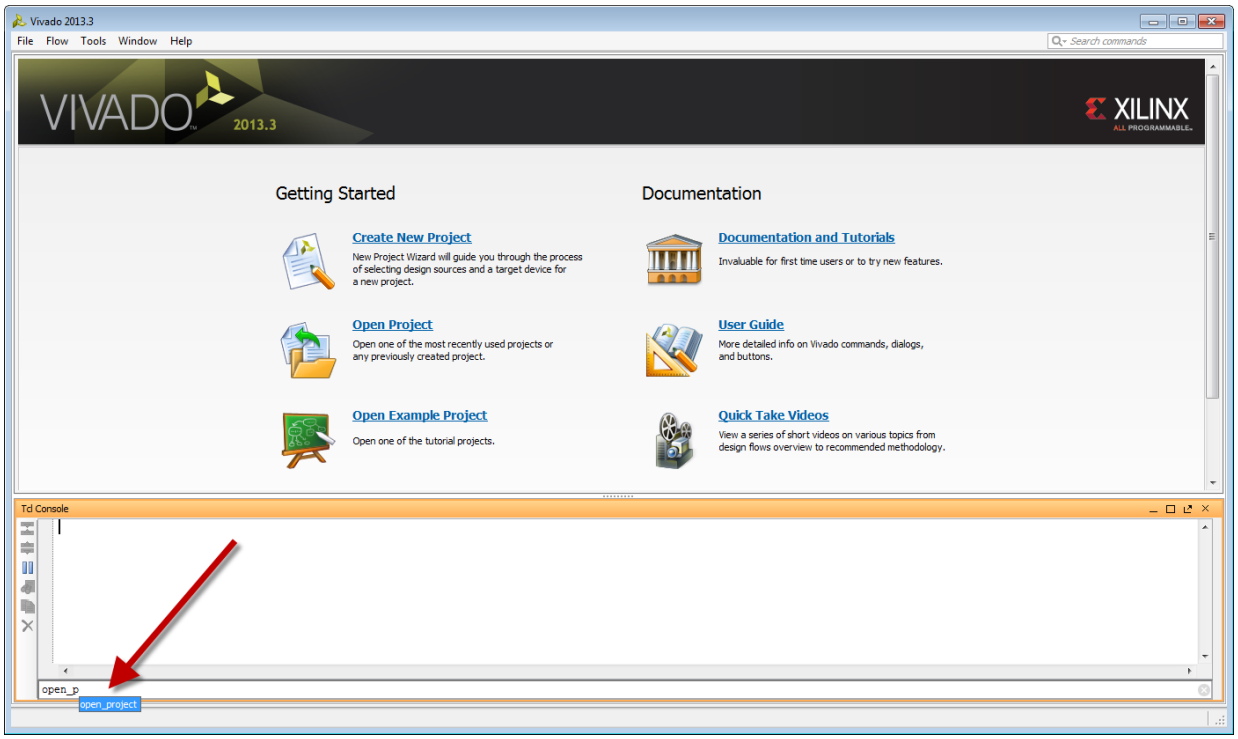


Figure 6 - Vivado TCL Console

6. Our project will open. Open the block design using the following TCL command:

open_bd_design ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd

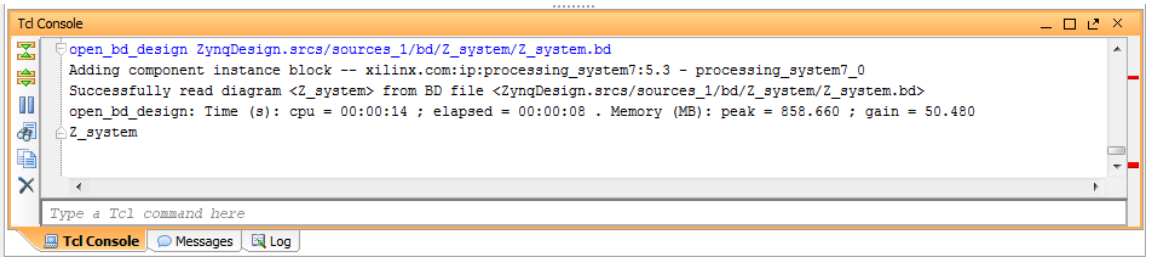


Figure 7 - Open Block Diagram

To recap, we've opened our project all the way down to the block design with only using TCL commands. One of the powerful features of TCL is scripting. And these commands can be stored and run from a single script.

With the block design open there are many things we can do with TCL. We can query a list of ports, create interface ports and nets, and even get a list of address spaces.

7. Start by querying a list of interfaces ports, type "**get_bd_intf_ports**". Of course our design is simple and only two ports are reported; DDR and FIXED_IO.

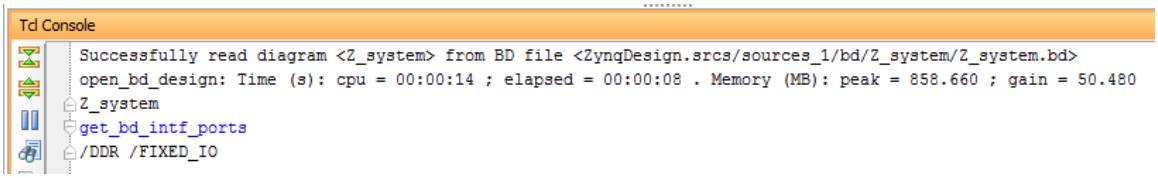
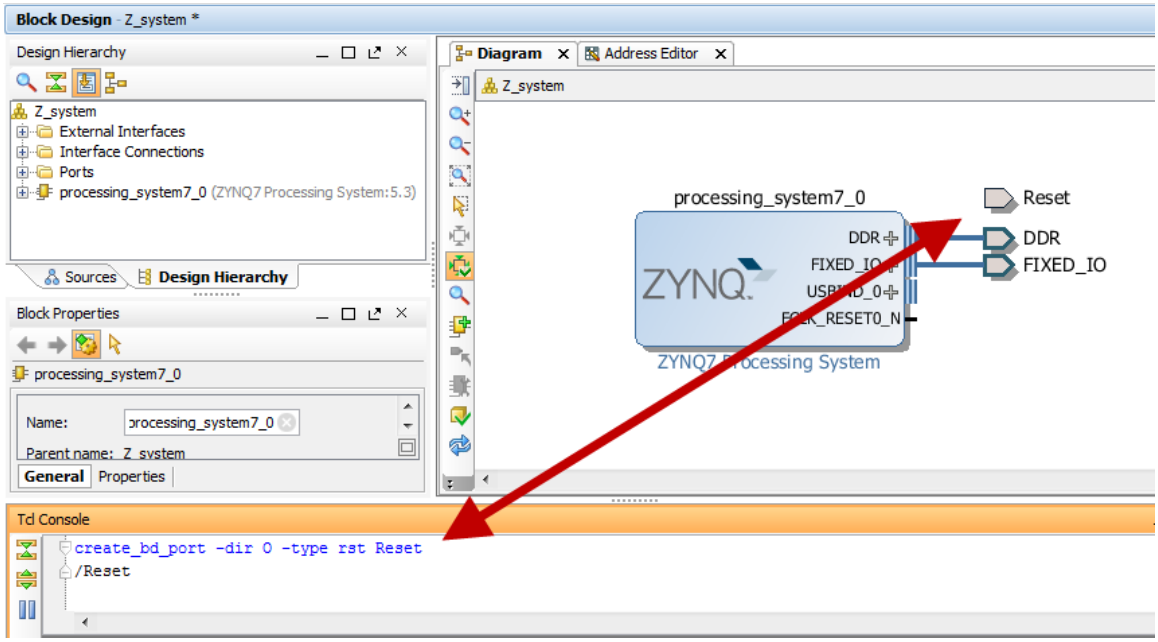


Figure 8 - Get Interface Ports

8. Ports can be created right from the TCL console. Create an output port for reset, type "**create_bd_port -dir O -type rst Reset**". Note: all created ports require a direction type [I | O | IO] and optionally a type [CLK | RST | CE | INTR | DATA | UNDEF].



9. Ports can be deleted as well. And if you are not sure what TCL command is being used to perform a task, Vivado echoes all GUI commands via the TCL Console. Right-click on the newly created Reset port and select **Delete**.

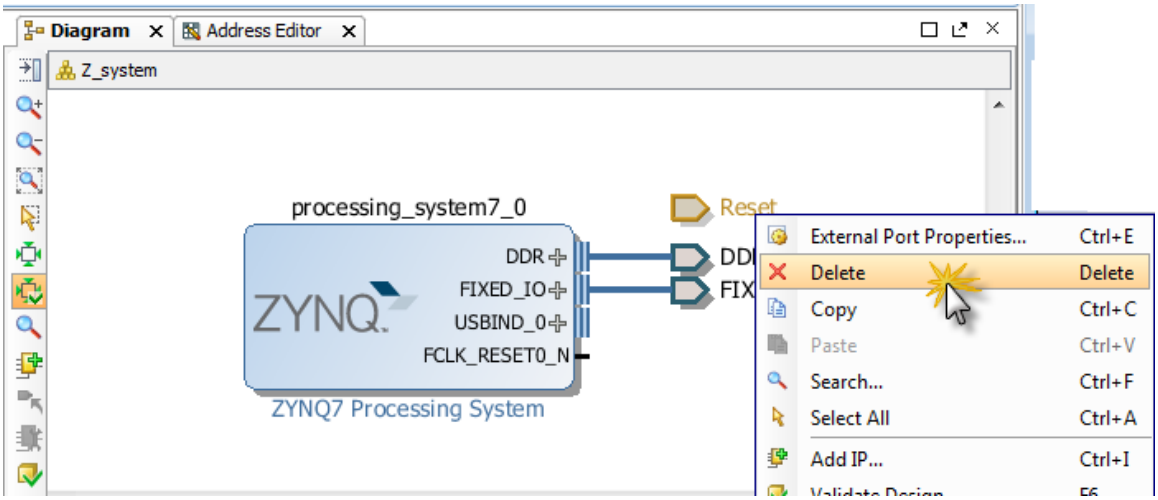


Figure 9 - Delete Ports

Notice in the TCL console, the GUI TCL command was echoed:

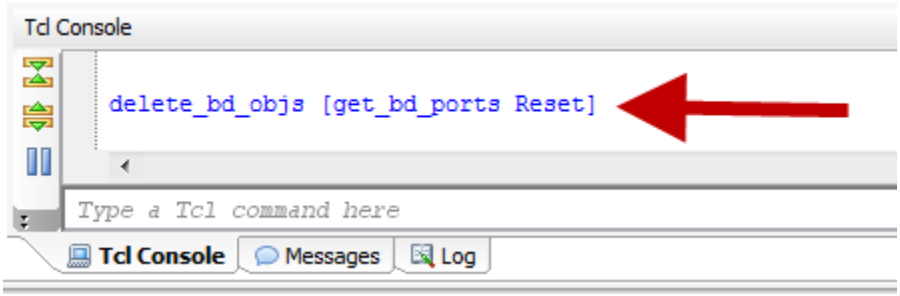


Figure 10 - TCL Console echoes GUI commands

Perhaps the most useful TCL command with regard to an IP Integrator based design is **write_bd_tcl**. This TCL command exports the current design to a TCL file on disk. Moreover, script files lets you recreate, reuse, and customize IP Integrator subsystem designs without having to archive the original subsystem design.

10. Run help on write_bd_tcl: **help write_bd_tcl**

write_bd_tcl

Description: Export the current design to a Tcl file on disk.

Syntax:
write_bd_tcl [-force] [-no_mig_contents] [-quiet] [-verbose] <name>

Returns:
TCL_OK, TCL_ERROR if failed.

Usage:

Name	*Description*
[-force]	*Flag to overwrite existing file.*
[-no_mig_contents]	*Flag to not include MIG PRJ contents into generated Tcl script, but instead will load PRJ from working directory. Default is to include MIG PRJ contents in Tcl script.*
[-quiet]	*Ignore command errors*
[-verbose]	*Suspend message limits during command execution*
<name>	*Name exported Tcl file*

Example:
The following example creates a Tcl script from the current IP Integrator subsystem design:

write_bd_tcl C:/Data/myDesign.tcl

Next, we'll export the block design using this command. Note, the block design not only encompasses all connections and IP in the design, but also the IP configurations. In this case, the only IP is our Zynq Processing System. But at this point it is a good idea to save this as it could be the starting point for any new designs.

11. Run the following TCL command: **write_bd_tcl basic_design.tcl**

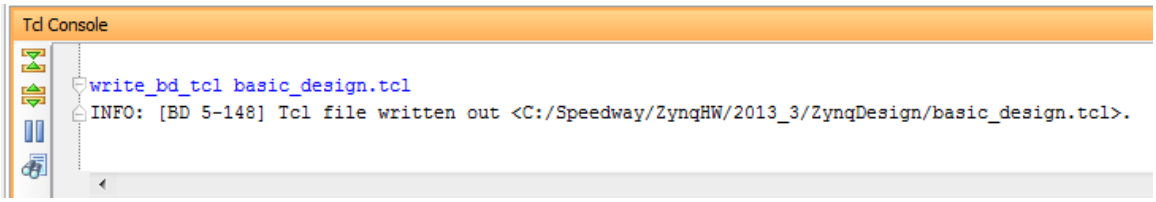


Figure 11 - Write out block design to TCL file

12. Using Window Explorer, **open this TCL file** with a text editor and browse through the script to see what was created.

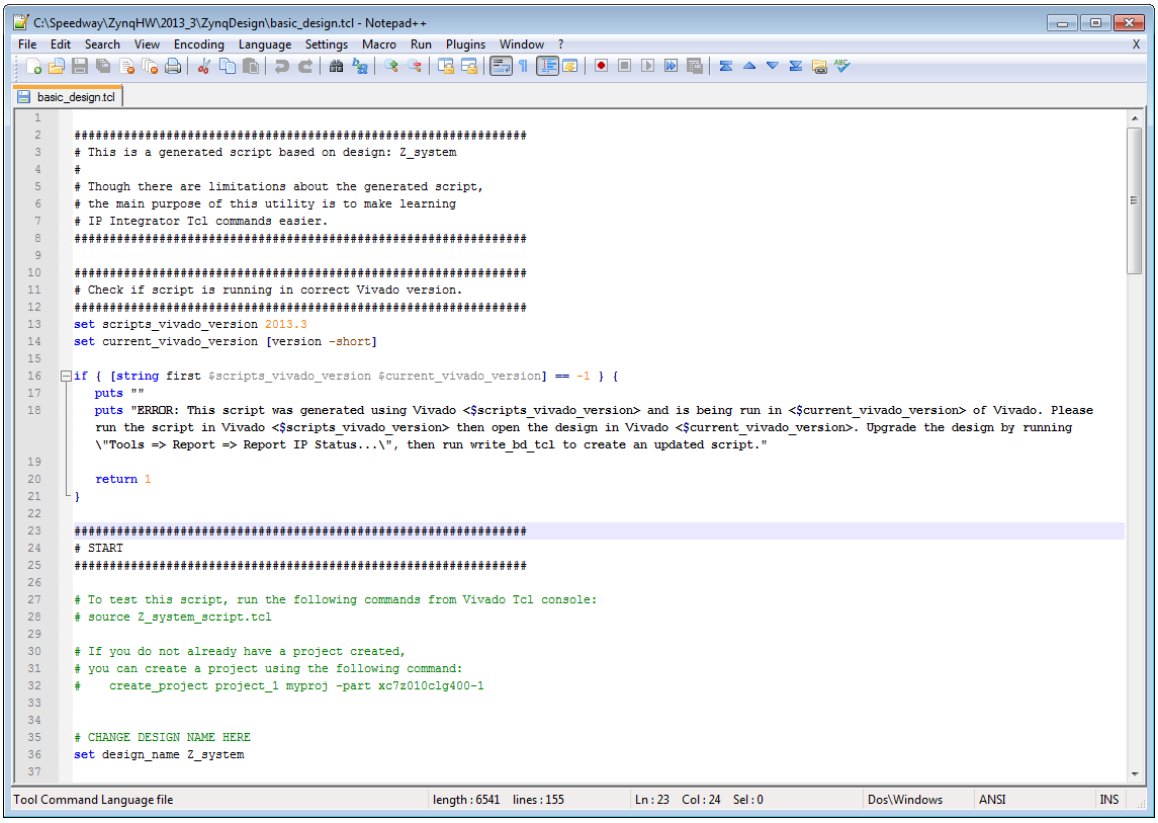


Figure 12 - Basic_design.tcl output

13. This encapsulates our block design, but not our Project settings. Another TCL file does that. Run **write_project_tcl project_setup.tcl**. Explore this file.

14. Read the warning about local sources. Look for the list of required sources.

15. **Close Vivado**, you do not need to save the design.

16. Keep the TCL files open in the text editor.

Questions:

Browse the basic_design.tcl file and answer the following questions:

- *What is the project name?*

_____

- *How many interface ports do you see? What are they?*

_____

- *What is the MIO Bank1 Voltage set to?*

_____

Browse the project_setup.tcl file and answer the following questions:

- *In what directory will the project be created?*

_____

- *What part is selected?*

_____

- *Does the TCL file load the block design?*

_____

- *Does the TCL file reload the synthesis and implementation settings?*

_____

- *In the project_setup.tcl script, what files does the script need to recreate the project?*

_____

Experiment 3: TCL Test

This experiment will test the TCL commands we've just explored. We'll put these commands together to create a script that recreates our project.

Experiment 3 General Instruction:

Modify the generated TCL scripts to recreate our project

Experiment 3 Step-by-Step Instructions:

1. Open Vivado.
2. In Vivado, Type **PWD** in the TCL Console. What is the current directory?
3. Change the directory to a temp directory. _____

   cd c:/Speedway/ZynqHW/2013_3/

   As we learned in the last experiment, the create_project.tcl file creates the project in the current directory. So we don't overwrite our current directory, create a new directory. We've used *cd* and *pwd* to change and print working directories, thus you'd expect a mkdir command, but it does not exist. Instead it is a function of the *file* command.

4. Type **help file** in the Vivado TCL Console.

   Scroll down and you'll see help on "file mkdir". This is the command we need to create a new directory.

5. Create a new directory running the following command: **file mkdir Temp**
6. Type "**cd ./**" and the TCL window will list all available directories:

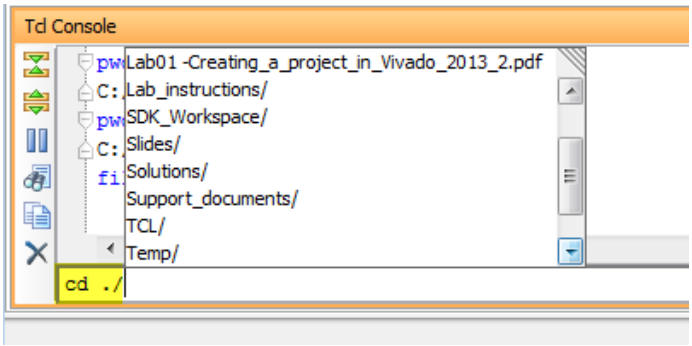


Figure 13 - Current Directory Structure

7. Click on the Temp/ directory from that menu list or enter the change directory command, **cd Temp.**

Now it's time to run the TCL script that will create our project.

8. Source the *project_setup.tcl*, which exists in the directory just above the Temp directory we just created.

   source ../ZynqDesign/project_setup.tcl

Vivado will open:

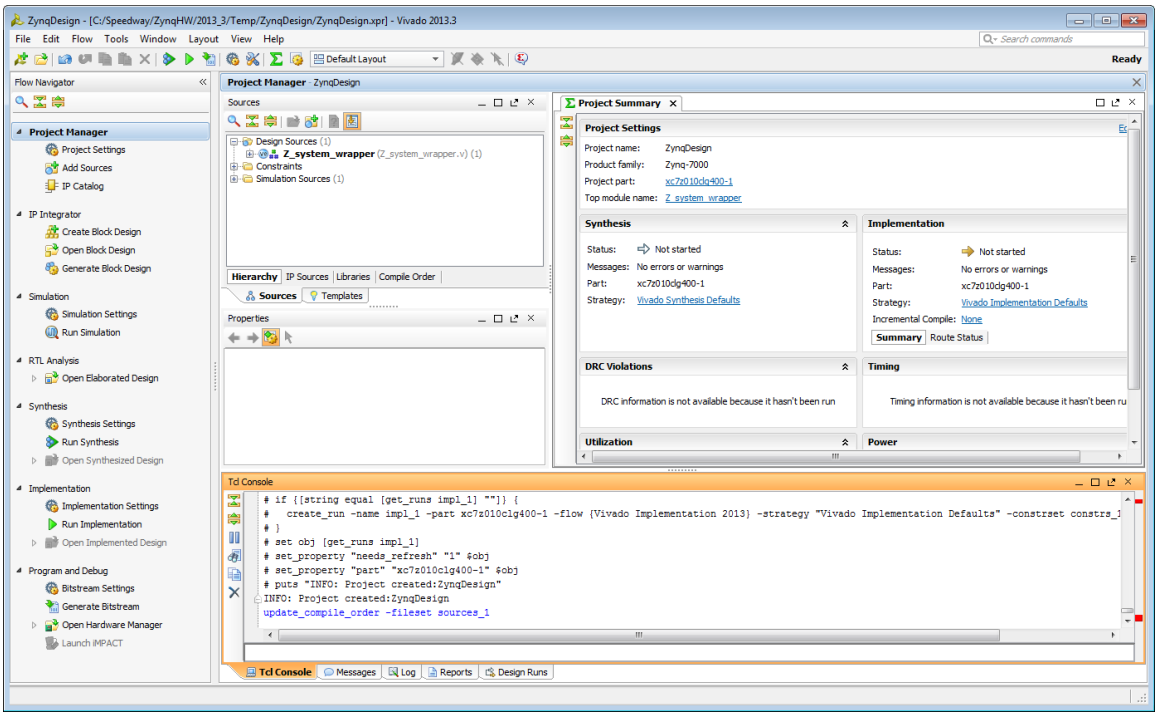


Figure 14 - Launching Vivado Project from TCL

Because we opened this on the same PC as the project was created on it was able to reference the Block Design from the original project directory as done in these TCL commands from *project_setup.tcl*:

```
# Add files to 'sources_1' fileset
set obj [get_filesets sources_1]
set files [list \
 "[file normalize "$orig_proj_dir/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd"]"\
 "[file normalize "$orig_proj_dir/ZynqDesign.srcs/sources_1/bd/Z_system/hdl/Z_system_wrapper.v"]"\
]
add_files -norecurse -fileset $obj $files

# Import local files from the original project
set files [list \
 "C:/Speedway/ZynqHW/2013_3/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd"\
 "C:/Speedway/ZynqHW/2013_3/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/hdl/Z_system_wrapper.v"\
]
set imported_files [import_files -fileset sources_1 $files]
```

Figure 15 - Project_setup.tcl imports block design from original project

But if this project were opened on another PC, those files would not be available. Thus we created a separate TCL script to create the block design, *basic_design.tcl*.

9. To show this, open the block design

   open_bd_design
   ./ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd

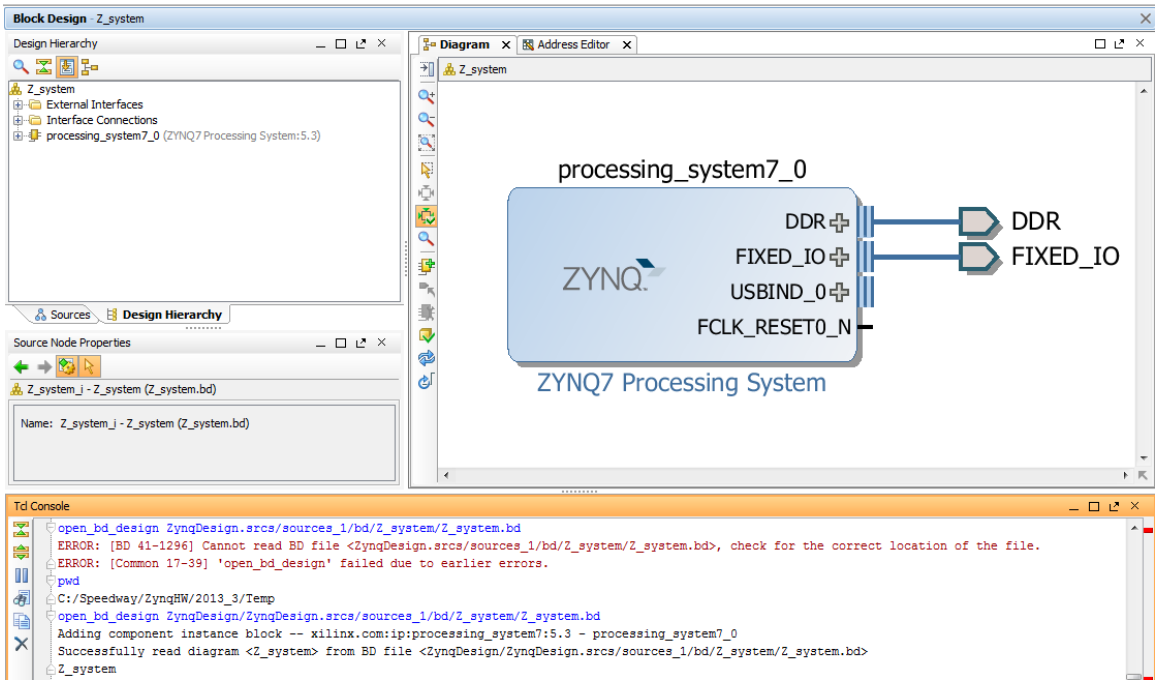


Figure 16 - Open Block Design

10. **Close** the block design by running:

    close_bd_design [get_bd_designs Z_system]

11. If asked, click **Yes** to save it.

12. Delete the block design by right-clicking on it in the sources view and clicking on **Remove File from Project…** Also check the box to **delete the project from disk** and click **OK**.

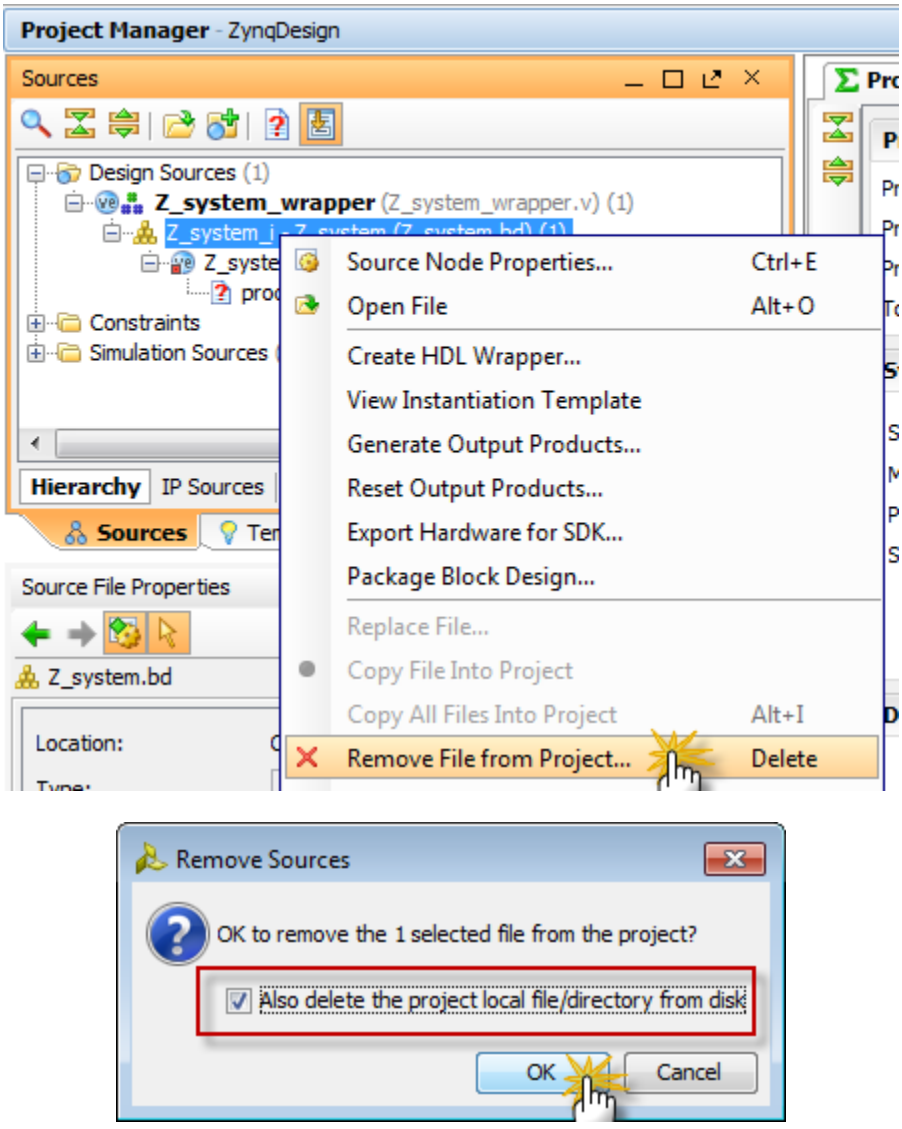


Figure 17 - Delete Block Design

Or, run the TCL commands:

file delete -force C:/Speedway/ZynqHW/2013_3/Temp/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system

remove_files C:/Speedway/ZynqHW/2013_3/Temp/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd

Now that the block design has been deleted and removed, it would be similar to use opening our project on a different computer.

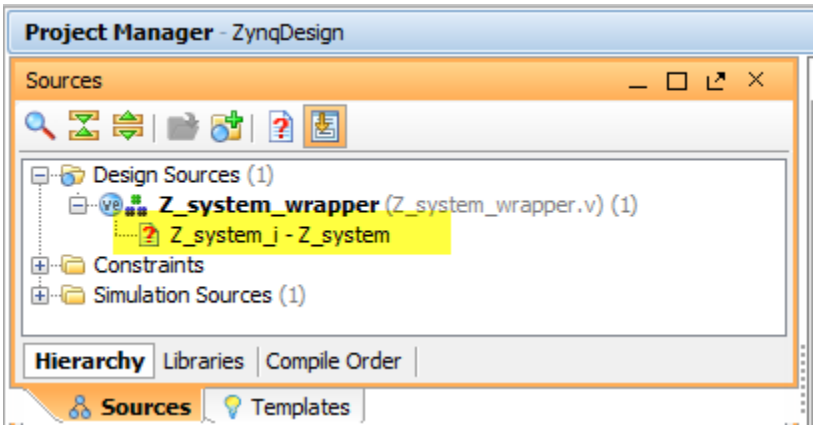


Figure 18 - Block Design Missing

13. Recreate the block design running the TCL script, **basic_design.tcl**.

    source ../ZynqDesign/basic_design.tcl

    The block design not only is recreated but it opens in the GUI. Open the Zynq IP block to verify it's the same. (Note: Click Regenerate Layout to see this view)

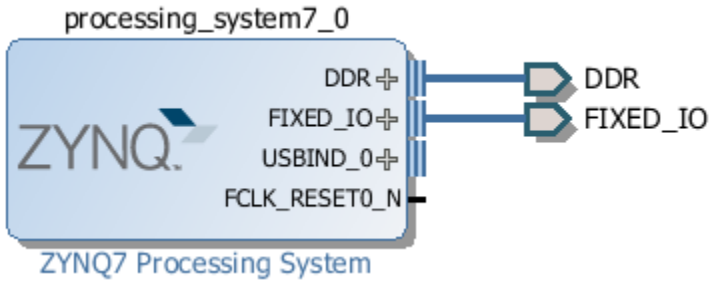


Figure 19 - Restored Block Design

14. Close the Vivado project. **close_project**

That concludes this lab. We know now a few basic TCL commands and how to enter them. Also how Vivado will create simple scripts that automatically restore our project!

Questions:

Browse the basic_design.tcl file and answer the following questions:

- *Does the Zynq IP block look the same? Are all I/O peripherals mapped the same?*

_____

- *Could both of the TCL files used to recreate this project be combined into one TCL file?*

_____

- *What source files were needed to recreate this entire project? In other words, what do you need to archive for revision control at this point?*

_____

Exploring Further

If you have more time and would like to investigate more…

- Create a single TCL file that recreates the entire project

This concludes Lab 4.

Revision History

Date	Version	Revision
6 Nov 13	02	Initial Draft

Resources

www.microzed.org

www.zedboard.org

www.xilinx.com/zynq

www.xilinx.com/sdk

www.xilinx.com/vivado

Answers

Experiment 2

- *What is the project name?*

 # CHANGE DESIGN NAME HERE
set design_name Z_system

- *How many interface ports do you see? What are they?*

2 – DDR & FIXED_IO

- *What is the MIO Bank1 Voltage set to?*

CONFIG.PCW_PRESET_BANK1_VOLTAGE {LVCMOS 1.8V}

- *In what directory will the project be created?*

This is a tough question. The first TCL command in this script sets the original project location, but that is only used to import source files. The second TCL command creates the project and does it in whatever directory the TCL script was sourced from. So before running this script, make sure you are in the directory you want the project created.

- *What part is selected?*

set_property "part" "xc7z010clg400-1" $obj

- *Does the TCL file load the block design?*

Yes it does:
 # Import local files from the original project
**set files [list **
"C:/Speedway/ZynqHW/2013_3/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd"

But it gets this source from our original project directory. So if run on another PC, the project will not get recreated correctly.

- *Does the TCL file reload the synthesis and implementation settings?*

Yes, see Create 'synth_1' run and Create 'impl_1' run

- *In the project_setup.tcl script, what files does the script need to recreate the project?*

ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd"
ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/hdl/Z_system_wrapper.v"

Experiment 3

- *Does the Zynq IP block look the same? Are all I/O peripherals mapped the same?*

Yes, it is the same

- *Could both of the TCL files used to recreate this project be combined into one TCL file?*

Yes, they could simply be combined, but the project_setup.tcl would need to remove importing the block design source. Otherwise when the basic_design.tcl is run, it will already see a block design with the same name and quit running.

- *What source files were needed to recreate this entire project? In other words, what do you need to archive for revision control at this point?*

None, only the two TCL scripts are required. Per the above question, one script could be created that performs the entire project creation, block design creation, and Zynq IP Customization. That's quite powerful. Thus only the TCL file would need to be checked into a revision control system. The system_wrapper HDL file is not required as it can recreated with a TCL command.