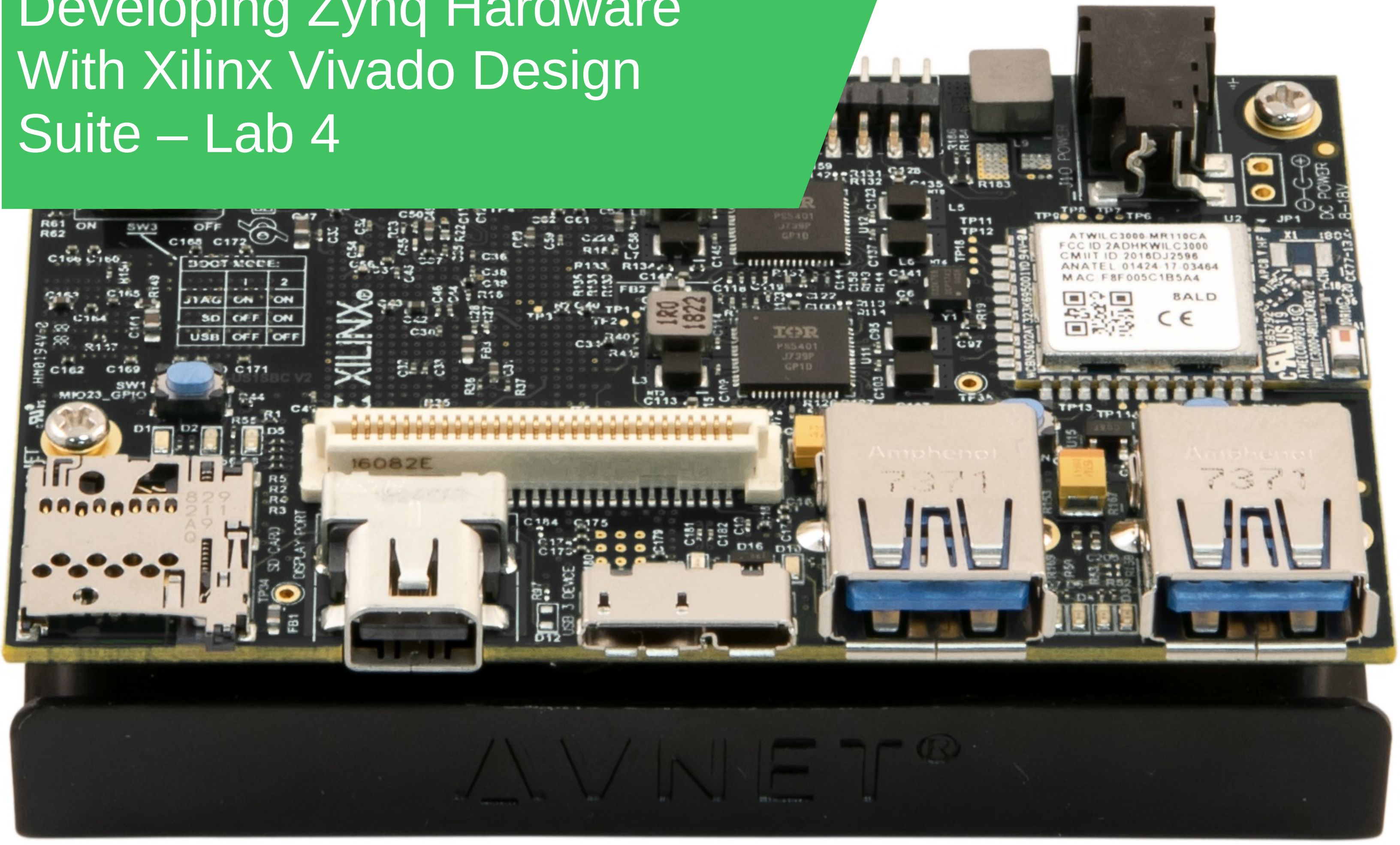
Avnet Technical Training Course

Developing Zynq Hardware With Xilinx Vivado Design Suite – Lab 4



Tools:	2019.1
Training Version:	v12
Date:	July 2019

Lab 4 Overview

This lab will give users a brief introduction to various Tool Command Language (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 a 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.

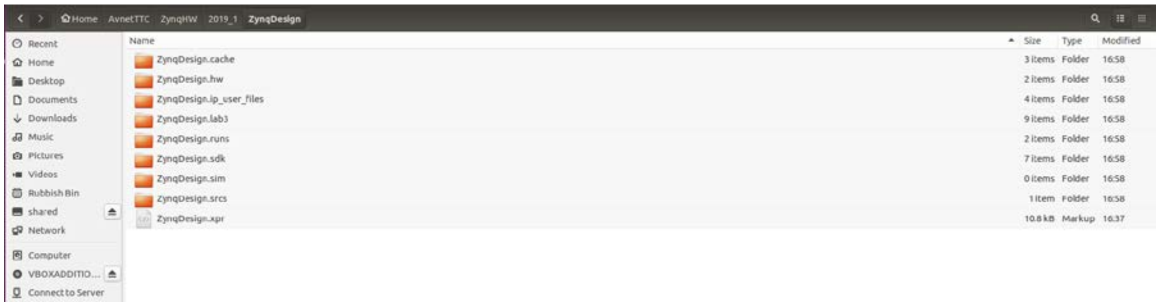
Experiment 1 Step-by-Step Instructions:

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/2019_1` folder. Then unzip **`Solutions_Minized\ZynqHW_Lab3_Solution.zip`** to the `2019_1` folder. If you have Archive Manager installed, you can do this by right-clicking and selecting Archive Manager then extracting in to the `2019_1` folder.

2. Launch **File Explorer**

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

   /home/training/AvnetTTC/\ZynqHW/2019_1/ZynqDesign



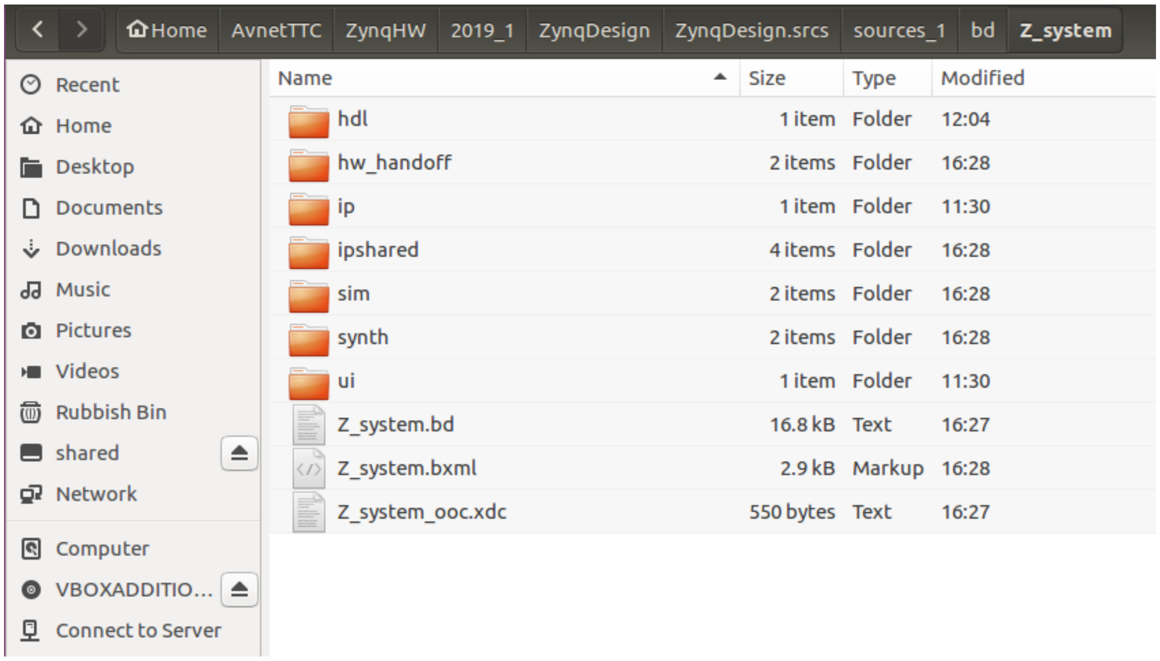
Name	Size	Type	Modified
ZynqDesign.cache	3 items	Folder	16:58
ZynqDesign.hw	2 items	Folder	16:58
ZynqDesign.ip_user_files	4 items	Folder	16:58
ZynqDesign.lab3	9 items	Folder	16:58
ZynqDesign.runs	2 items	Folder	16:58
ZynqDesign.sdk	7 items	Folder	16:58
ZynqDesign.sim	0 items	Folder	16:58
ZynqDesign.srcs	1 item	Folder	16:58
ZynqDesign.xpr	10.8 kB	Markup	16:37

Figure 1 - Project Directory Contents

4. There is only one Vivado 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

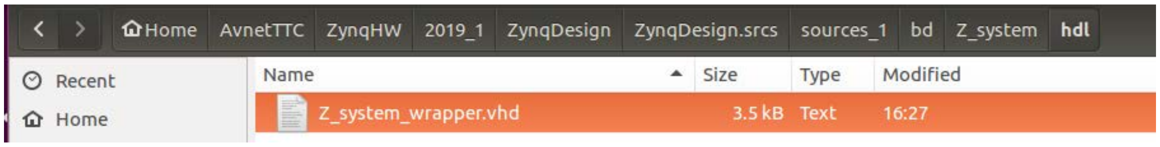

Home / AvnetTTC / ZynqHW / 2019_1 / ZynqDesign / ZynqDesign.srcs / sources_1 / bd / Z_system

Name	Size	Type	Modified
hdl	1 item	Folder	12:04
hw_handoff	2 items	Folder	16:28
ip	1 item	Folder	11:30
ipshared	4 items	Folder	16:28
sim	2 items	Folder	16:28
synth	2 items	Folder	16:28
ui	1 item	Folder	11:30
Z_system.bd	16.8 kB	Text	16:27
Z_system.bxml	2.9 kB	Markup	16:28
Z_system_ooc.xdc	550 bytes	Text	16:27

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.


Home / AvnetTTC / ZynqHW / 2019_1 / ZynqDesign / ZynqDesign.srcs / sources_1 / bd / Z_system / hdl

Name	Size	Type	Modified
Z_system_wrapper.vhd	3.5 kB	Text	16:27

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.

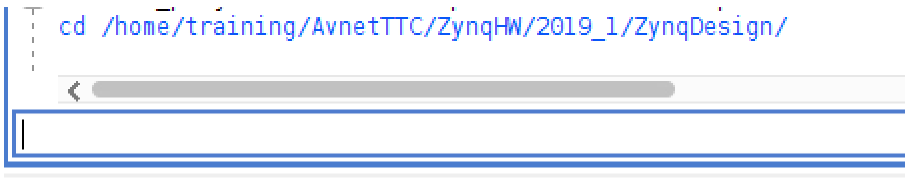
Experiment 2 Step-by-Step Instructions:

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. Click the **Tcl Console** tab at the bottom of the Vivado initial window and 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.)

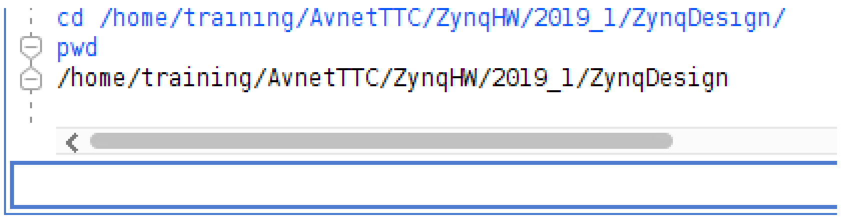
   cd /home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign



```
cd /home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign/
```

Figure 4 - Change Working Directory

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

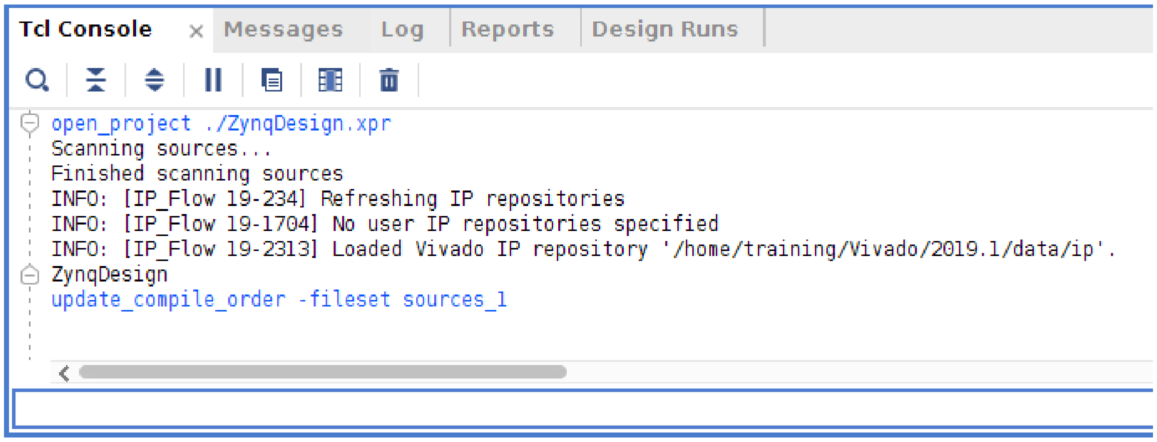


```
cd /home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign/
pwd
/home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign
```

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 the relative path to our Vivado project name. By starting the path with "./" this will activate the auto-complete function for the path string.

open_project ./ZynqDesign.xpr

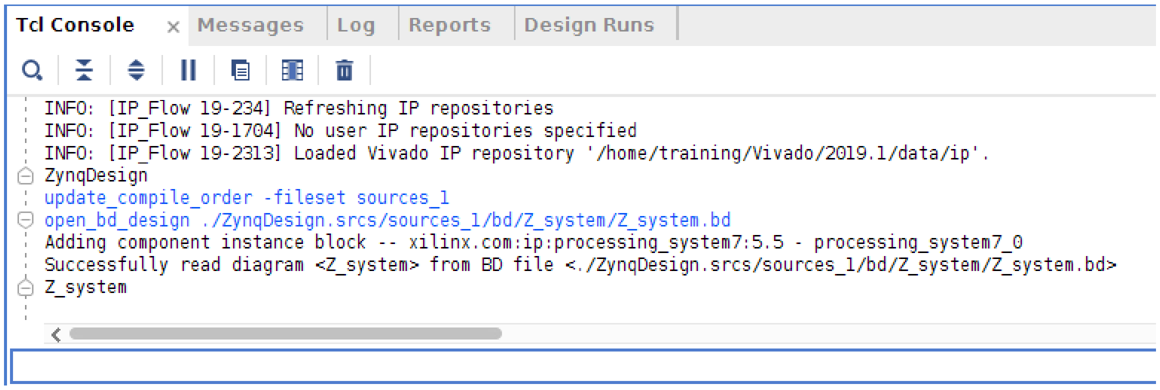


```
open_project ./ZynqDesign.xpr
Scanning sources...
Finished scanning sources
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository '/home/training/Vivado/2019.1/data/ip'.
ZynqDesign
update_compile_order -fileset sources_1
```

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



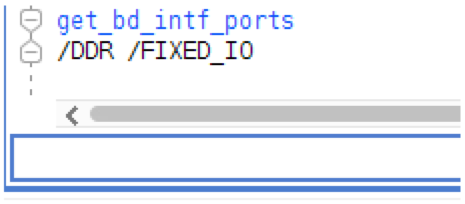
```
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository '/home/training/Vivado/2019.1/data/ip'.
ZynqDesign
update_compile_order -fileset sources_1
open_bd_design ./ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd
Adding component instance block -- xilinx.com:ip:processing_system7:5.5 - processing_system7_0
Successfully read diagram <Z_system> from BD file <./ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd>
Z_system
```

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.

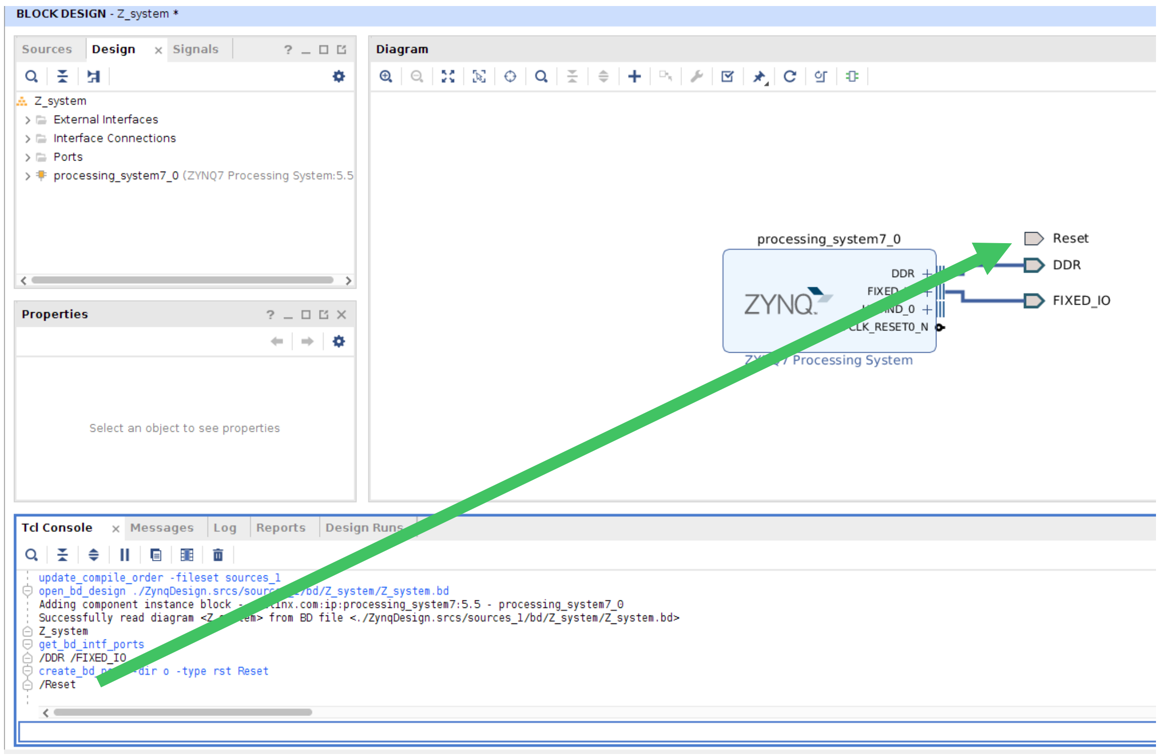
7. Start by querying a list of interfaces ports, type “**get_bd_intf_ports**”. Of course our design is simple and only two ports are reported; DDR and FIXED_IO.



```
get_bd_intf_ports
/DDR /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].



```
update_compile_order -fileset sources_1
open_bd_design ./ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd
Adding component instance block -- xilinx.com:ip:processing_system7:5.5 - processing_system7_0
Successfully read diagram <Z_system> from BD file <./ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd>
Z_system
get_bd_intf_ports
/DDR /FIXED_IO
create_bd_port -dir o -type rst Reset
/Reset
```

Figure 9 - Create Ports

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**.

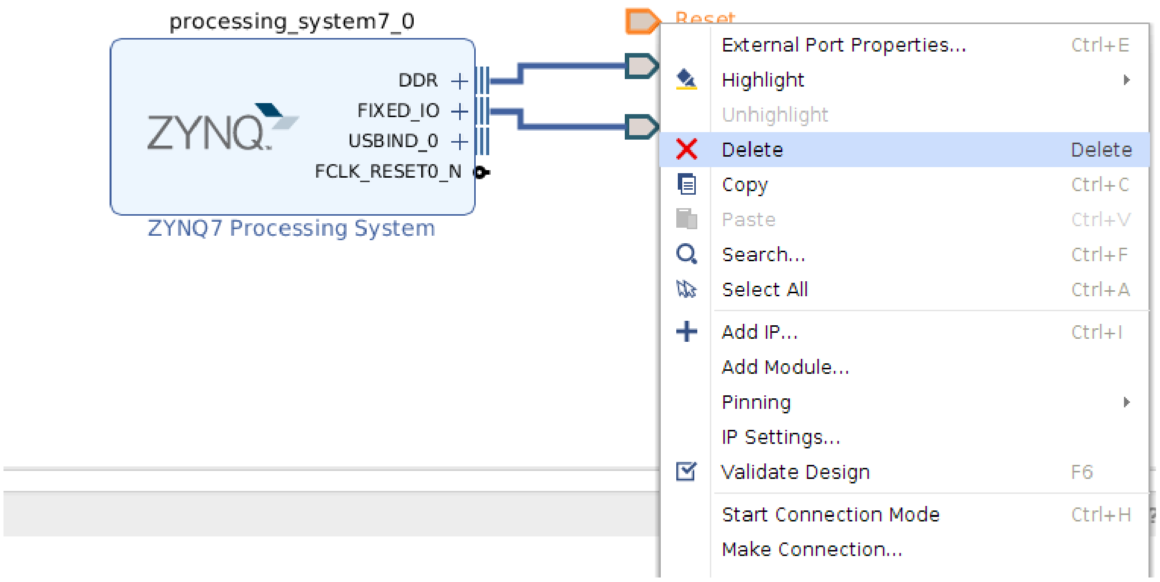
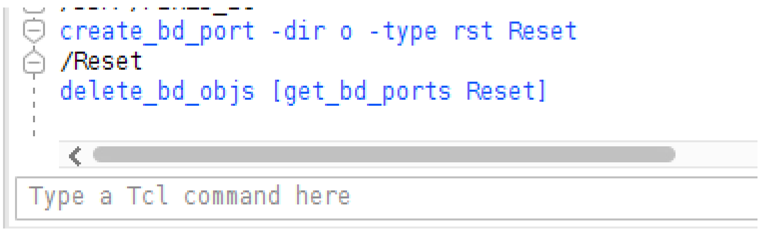


Figure 10 - Delete Ports

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

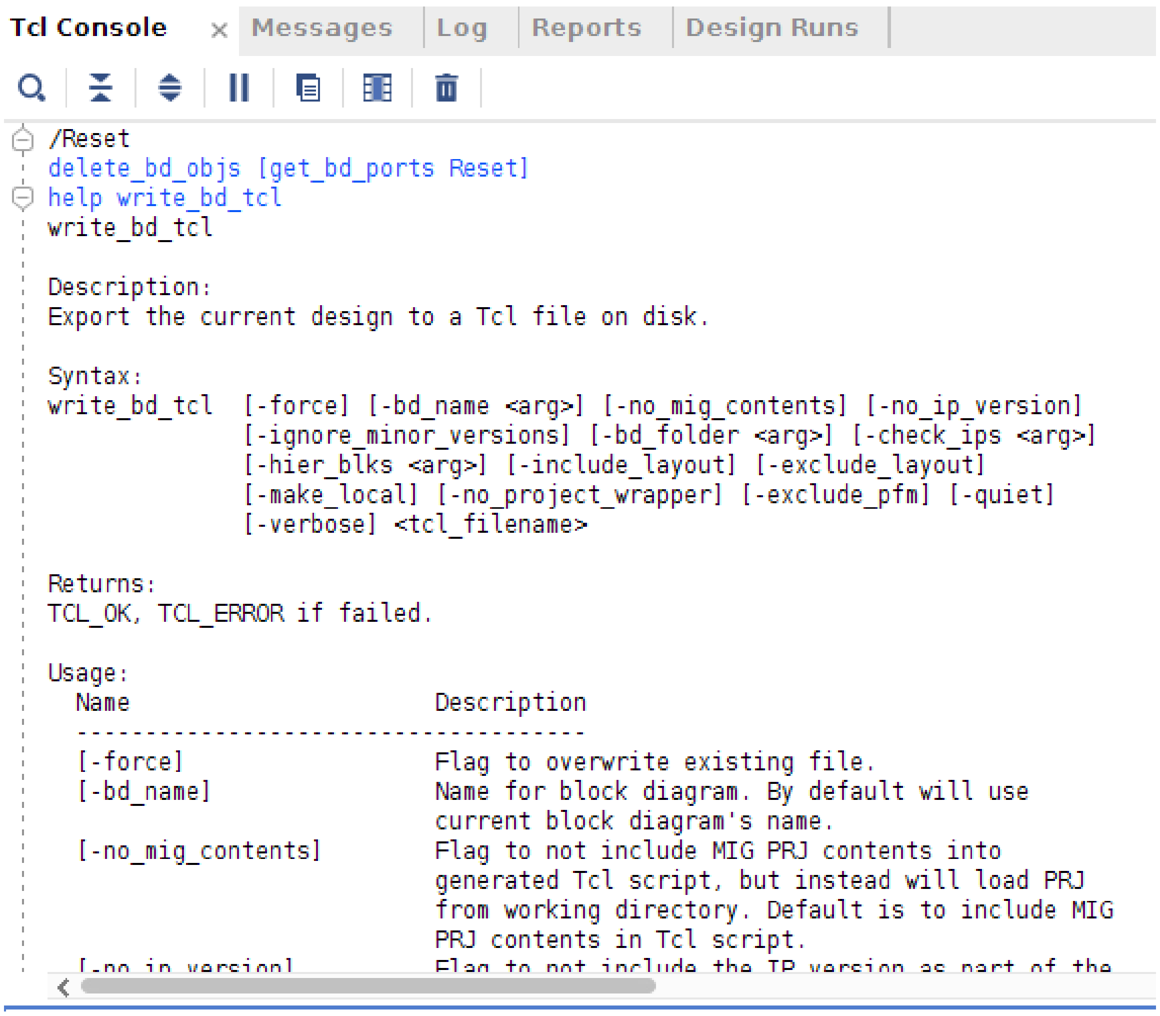


```
create_bd_port -dir o -type rst Reset
/Reset
delete_bd_objs [get_bd_ports Reset]
```

Figure 11 - 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 block design. 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**

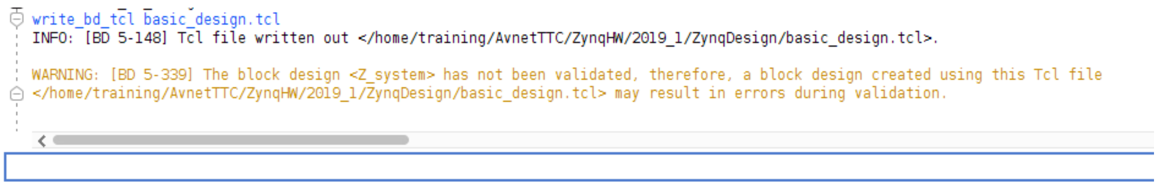


```
/Reset
delete_bd_objs [get_bd_ports Reset]
help write_bd_tcl
write_bd_tcl

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

Syntax:
write_bd_tcl  [-force] [-bd_name <arg>] [-no_mig_contents] [-no_ip_version]
              [-ignore_minor_versions] [-bd_folder <arg>] [-check_ips <arg>]
              [-hier_blks <arg>] [-include_layout] [-exclude_layout]
              [-make_local] [-no_project_wrapper] [-exclude_pfm] [-quiet]
              [-verbose] <tcl_filename>

Returns:
TCL_OK, TCL_ERROR if failed.

Usage:
  Name                          Description
  -------------------------------------
  [-force]                      Flag to overwrite existing file.
  [-bd_name]                    Name for block diagram. By default will use
                                current block diagram's name.
  [-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.
  [-no_ip_version]              Flag to not include the IP version as part of the
```

Figure 12 - TCL Console: help write_bd_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.

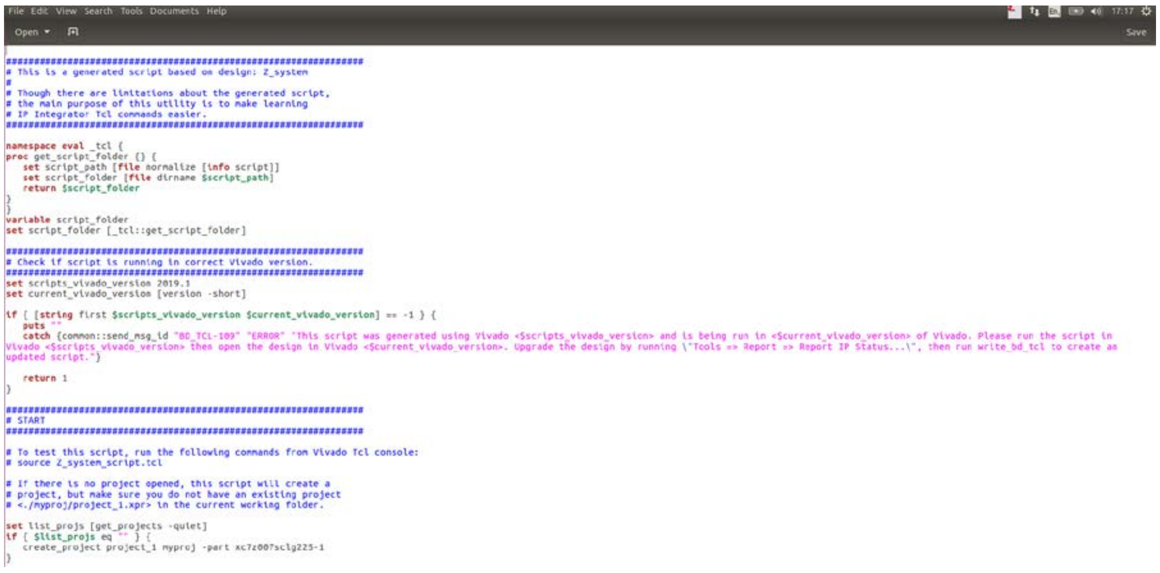
11. Run the following TCL command: **write_bd_tcl basic_design.tcl**



```
write_bd_tcl basic_design.tcl
INFO: [BD 5-148] Tcl file written out </home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign/basic_design.tcl>.

WARNING: [BD 5-339] The block design <Z_system> has not been validated, therefore, a block design created using this Tcl file
</home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign/basic_design.tcl> may result in errors during validation.
```

Figure 13 - Write out block design to TCL file

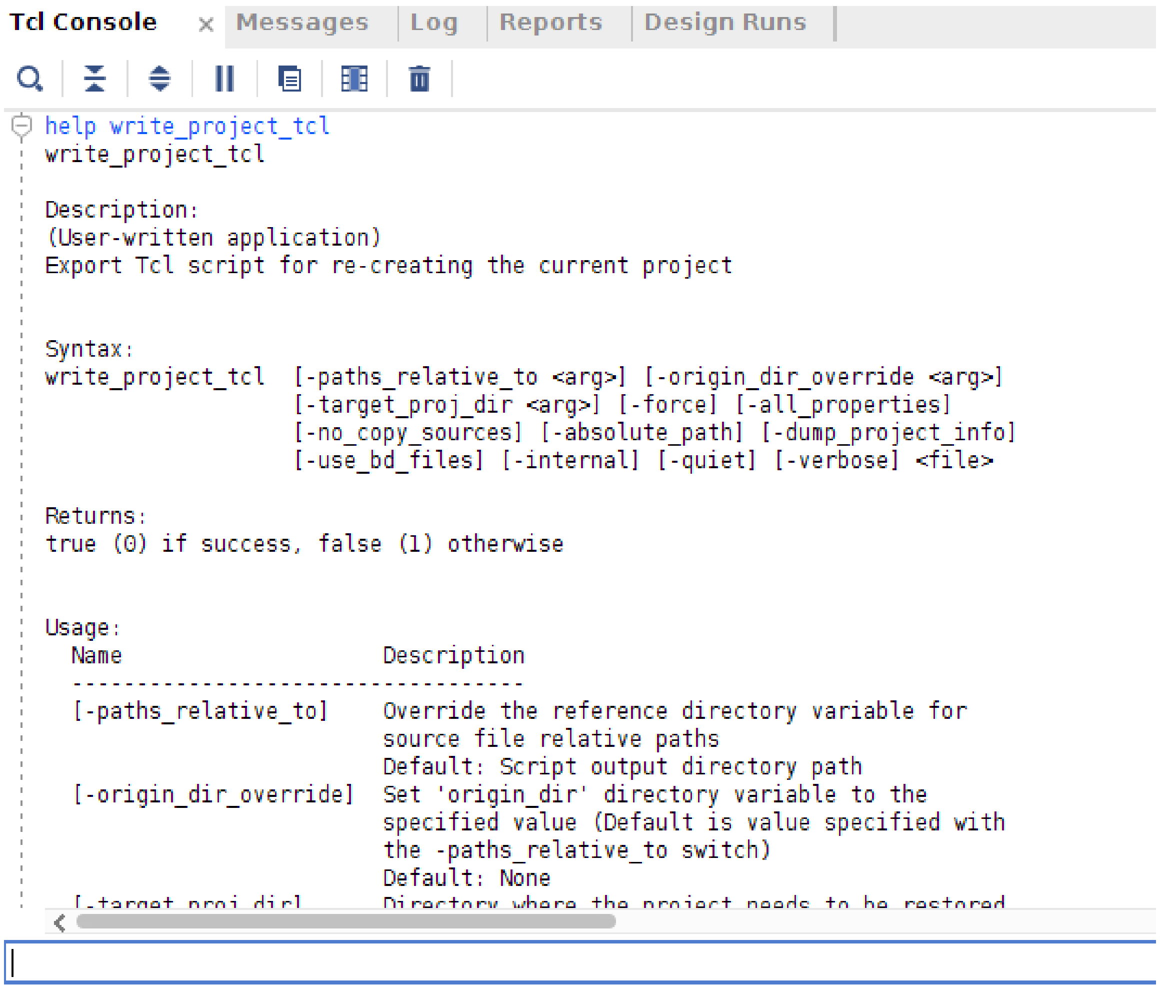
12. Using **gedit** or similar, **open this TCL file** with a text editor and browse through the script to see what was created.



```
################################################################
# This is a generated script based on design: Z_system
#
# Though there are limitations about the generated script,
# the main purpose of this utility is to make learning
# IP Integrator Tcl commands easier.
################################################################

namespace eval _tcl {
proc get_script_folder {} {
   set script_path [file normalize [info script]]
   set script_folder [file dirname $script_path]
   return $script_folder
}
}
variable script_folder
set script_folder [_tcl::get_script_folder]

################################################################
# Check if script is running in correct Vivado version.
################################################################
set scripts_vivado_version 2019.1
set current_vivado_version [version -short]

if { [string first $scripts_vivado_version $current_vivado_version] == -1 } {
   puts ""
   catch {common::send_msg_id "BD_TCL-109" "ERROR" "This script was generated using Vivado <$scripts_vivado_version> and is being run in <$current_vivado_version> of Vivado. Please run the script in
Vivado <$scripts_vivado_version> then open the design in Vivado <$current_vivado_version>. Upgrade the design by running \"Tools => Report => Report IP Status...\", then run write_bd_tcl to create an
updated script."}

   return 1
}

################################################################
# START
################################################################

# To test this script, run the following commands from Vivado Tcl console:
# source Z_system_script.tcl

# If there is no project opened, this script will create a
# project, but make sure you do not have an existing project
# <./myproj/project_1.xpr> in the current working folder.

set list_projs [get_projects -quiet]
if { $list_projs eq "" } {
   create_project project_1 myproj -part xc7z007sclg225-1
}
```

Figure 14 - basic_design.tcl output

13. This encapsulates our block design, but not our Project settings. Another TCL script, write_project_tcl, does that. Run help on write_project_tcl: **help write_project_tcl**.



```
help write_project_tcl
write_project_tcl

Description:
(User-written application)
Export Tcl script for re-creating the current project


Syntax:
write_project_tcl  [-paths_relative_to <arg>] [-origin_dir_override <arg>]
                   [-target_proj_dir <arg>] [-force] [-all_properties]
                   [-no_copy_sources] [-absolute_path] [-dump_project_info]
                   [-use_bd_files] [-internal] [-quiet] [-verbose] <file>

Returns:
true (0) if success, false (1) otherwise


Usage:
  Name                     Description
  -----------------------------------
  [-paths_relative_to]     Override the reference directory variable for
                           source file relative paths
                           Default: Script output directory path
  [-origin_dir_override]   Set 'origin_dir' directory variable to the
                           specified value (Default is value specified with
                           the -paths_relative_to switch)
                           Default: None
```

Figure 15 - TCL Console: help write_project_tcl

14. Run the following command to export the project settings:

    write_project_tcl -paths_relative_to /home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign project_setup.tcl

15. Open the **project_setup.tcl** file with a text editor and browse through the script to see what was created.

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

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

18. 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** into the TCL Console. What is the current directory?

3. Change the directory to **/home/training/AvnetTTC/ZynqHW/2019_1/**

cd /home/training/AvnetTTC/ZynqHW/2019_1/

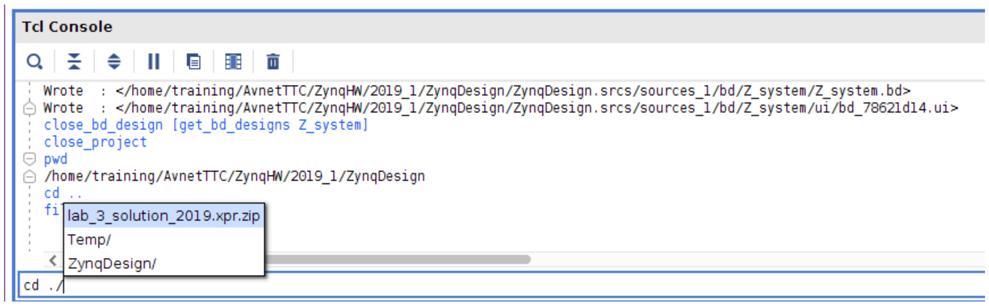
a. As we learned in the last experiment, the create_project.tcl file creates the project in the current directory. So we don't overwrite our current directory, create a new directory. We've used ***cd*** and ***pwd*** to change and print working directories, thus you'd expect a ***mkdir*** command, but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

4. Type ***help file*** and enter it into 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:



```
Wrote  : </home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd>
Wrote  : </home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/ui/bd_78621d14.ui>
close_bd_design [get_bd_designs Z_system]
close_project
pwd
/home/training/AvnetTTC/ZynqHW/2019_1/ZynqDesign
cd ..
fi
```

```
lab_3_solution_2019.xpr.zip
Temp/
ZynqDesign/
```

```
cd ./
```

Figure 16 - Current Directory Structure

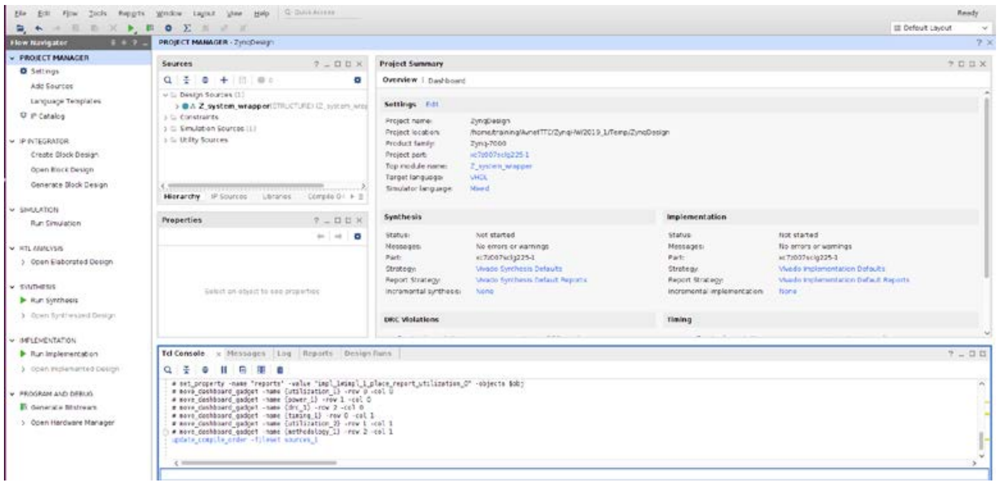
7. Click on the Temp/ directory from that menu list or enter the change directory command, **cd Temp.**

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

The Vivado Project Manager will open:



Project Summary

Overview | Dashboard

Settings

Project name:	ZynqDesign
Project location:	/home/training/AvnetTTC/Zynq-HW2019_1/Temp/ZynqDesign
Product family:	Zynq-7000
Project part:	xc7z007sclg225-1
Top module name:	Z_system_wrapper
Target language:	VHDL
Simulator language:	Mixed

Synthesis

Status:	Not started
Messages:	No errors or warnings
Part:	xc7z007sclg225-1
Strategy:	Vivado Synthesis Defaults
Report Strategy:	Vivado Synthesis Default Reports
Incremental synthesis:	None

Implementation

Status:	Not started
Messages:	No errors or warnings
Part:	xc7z007sclg225-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

Tcl Console

```
# set_property -name "reports" -value "impl_1#impl_1_place_report_utilization_0" -objects $obj
# move_dashboard_gadget -name {utilization_1} -row 0 -col 0
# move_dashboard_gadget -name {power_1} -row 1 -col 0
# move_dashboard_gadget -name {drc_1} -row 2 -col 0
# move_dashboard_gadget -name {timing_1} -row 0 -col 1
# move_dashboard_gadget -name {utilization_2} -row 1 -col 1
# move_dashboard_gadget -name {methodology_1} -row 2 -col 1
update_compile_order -fileset sources_1
```

Figure 17 - 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*:

```
# Set 'sources_1' fileset object
set obj [get_filesets sources_1]
# Import local files from the original project
set files [list \
 "[file normalize "$origin_dir/ZynqDesign.srcs/sources_1/bd/Z_system/Z_system.bd"]"\
 "[file normalize "$origin_dir/ZynqDesign.srcs/sources_1/bd/Z_system/hdl/Z_system_wrapper.v"]"\
]
set imported_files [import_files -fileset sources_1 $files]
```

Figure 18 - 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

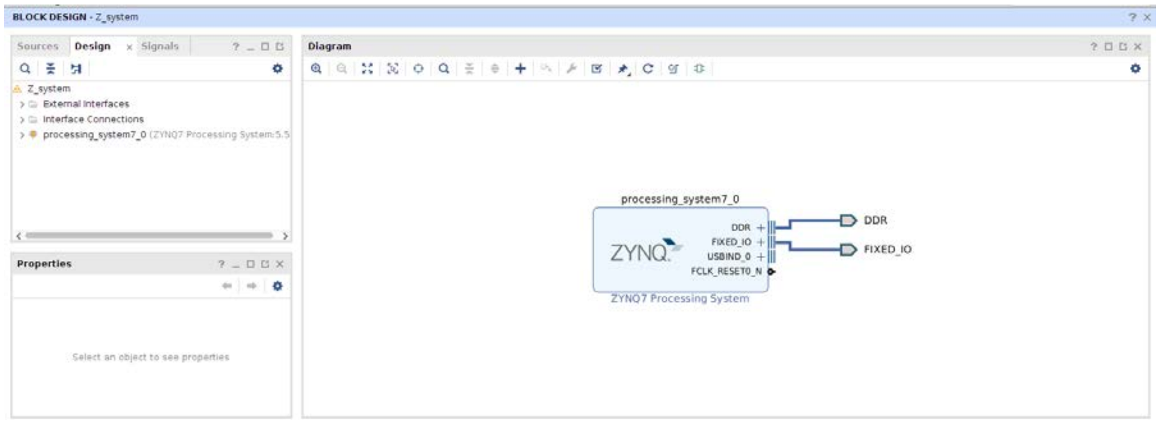


Figure 19 - Open Block Design

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…** Then check the **Also delete the project local file/directory from check box** and click **OK**.

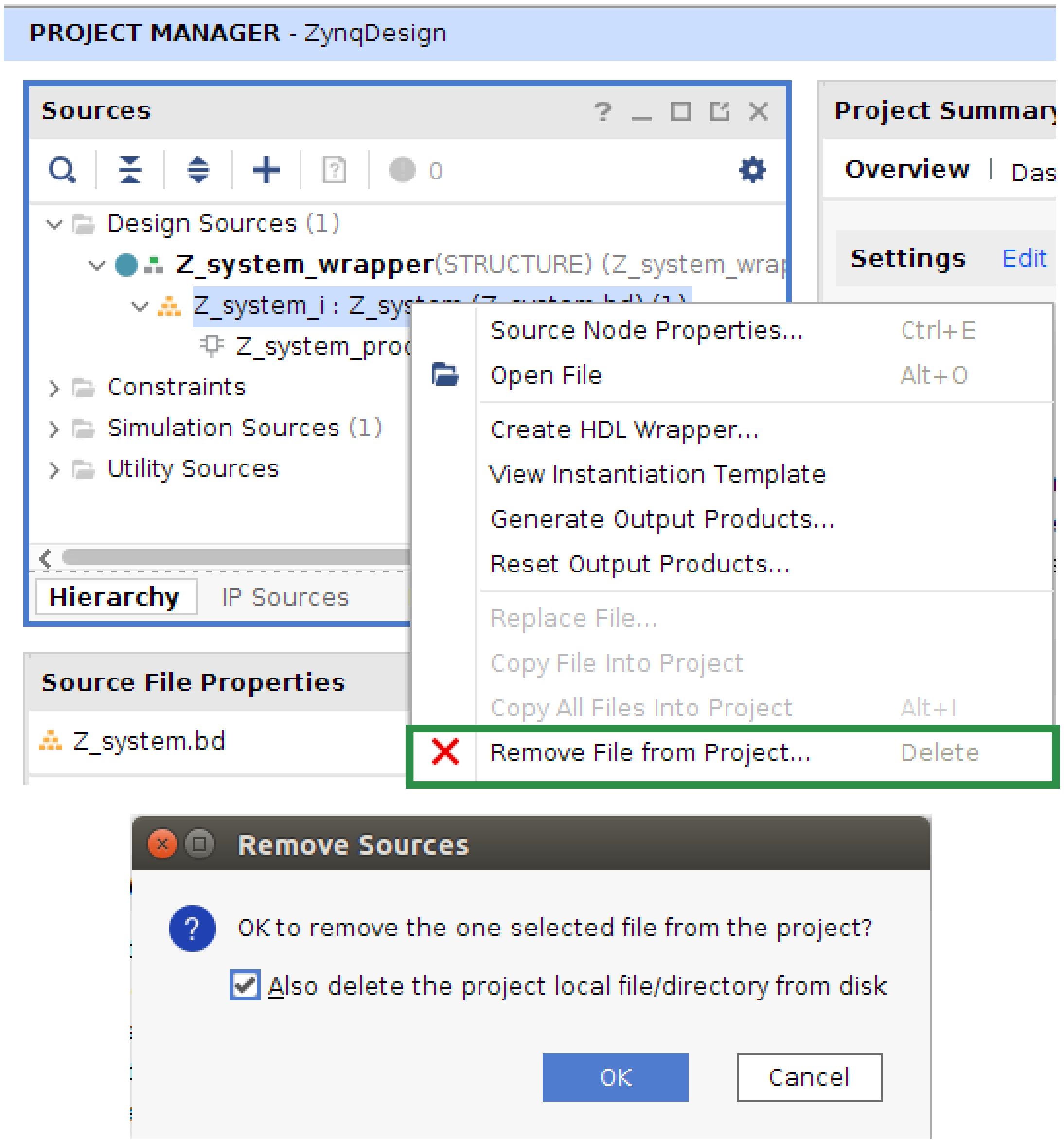

PROJECT MANAGER - ZynqDesign
Sources
Design Sources (1)
Z_system_wrapper(STRUCTURE) (Z_system_wrap
Z_system_i : Z_system (Z_system.bd) (1)
Z_system_proc
Constraints
Simulation Sources (1)
Utility Sources
Hierarchy
IP Sources
Source File Properties
Z_system.bd
Project Summary
Overview
Das
Settings
Edit
Source Node Properties... Ctrl+E
Open File Alt+O
Create HDL Wrapper...
View Instantiation Template
Generate Output Products...
Reset Output Products...
Replace File...
Copy File Into Project
Copy All Files Into Project Alt+I
Remove File from Project... Delete
Remove Sources
OK to remove the one selected file from the project?
Also delete the project local file/directory from disk
OK
Cancel

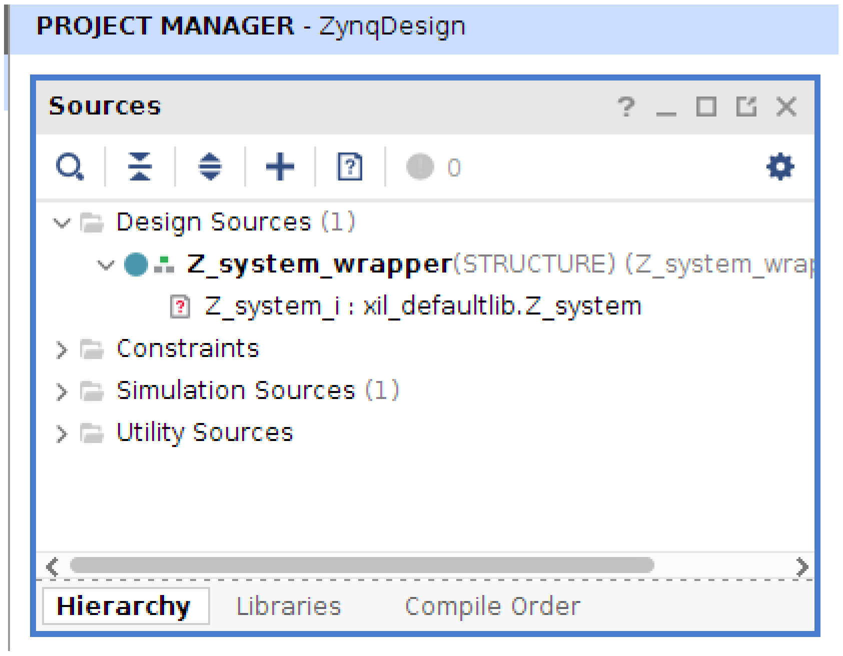
Figure 20 - Delete Block Design

Or, run the TCL commands:

```
file delete -force \
/home/training/AvnetTTC/ZynqHW/2019_1/Temp/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system
```

```
remove_files \
/home/training/AvnetTTC
/ZynqHW/2019_1/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 opening our project on a different computer.


PROJECT MANAGER - ZynqDesign

Sources

- Design Sources (1)
  - Z_system_wrapper(STRUCTURE) (Z_system_wrap
    - Z_system_i : xil_defaultlib.Z_system
- Constraints
- Simulation Sources (1)
- Utility Sources

Figure 21 - 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 also opens in the GUI. Open the Zynq IP block to verify it's the same. (Note: Click Regenerate Layout to see this view)

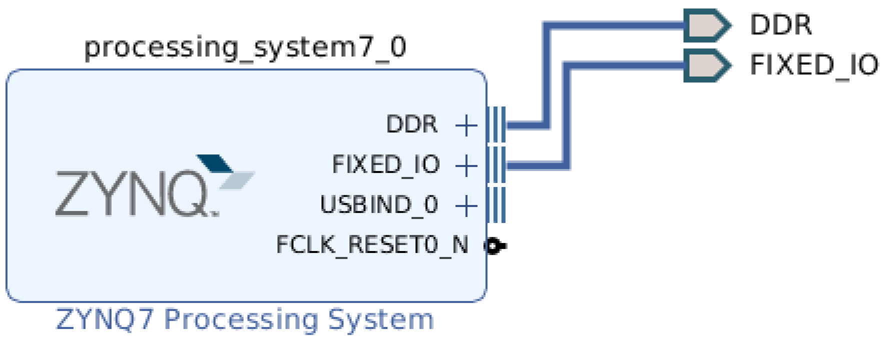


Figure 22 - Restored Block Design

14. Close the Vivado project using the TCL command **close_project**.

15. Close Vivado

16. The /home/training/AvnetTTC/ZynqHW/2019_1\Temp directory can be deleted.

That concludes this lab. We now know a few basic TCL commands, how to enter them, and 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
30 Oct 14	03	Updated for Vivado 2014.3
5 Jan 15	04	Updated for Vivado 2014.4
04 Mar 15	05	Finalized for Vivado 2014.4
17 Mar 15	06	Minor edits for release
Oct 2015	07	Updated to Vivado 2015.2
July 2016	08	Updated to Vivado 2016.2
May 2017	09	Updated to Vivado 2017.1
June 2017	10	Udated to Vivado 2017.1 for MiniZed + Rebranding
Jan 2018	11	Updated to Vivado 2017.4
July 2019	12	Updated to Vivado 2019.1

Resources

www.minized.org

www.microzed.org

www.picozed.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
Variable design_name
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?*

LVCMOS 3.3V

- *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 -name "part" -value "xc7z007sclg225-1" -objects $obj

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

Yes it does:

```
# Import local files from the original project
set files [list \
 "[file normalize
"$origin_dir/ZynqDesign.srcs/sources_1/bd/Z_system/hdl/Z_system_wrapper.vhd"]"\
```

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?*

```
C:/Speedway/ZynqHW/2017_4/ZynqDesign/ZynqDesign.srcs/sources_1/bd/Z_system/hdl
/Z_system_wrapper.vhd
```

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.