BlackParrot FPGA Simulation Guide

Vivado Setup

Before you run Vivado on KK9 (or other similar machine), make sure you source the settings file by running:

```
source /gro/cad/Xilinx/Vivado/2019.1/.settings64-Vivado.sh
```

To launch vivado, simply run the following command in the terminal

```
vivado &
```

(Optional) If you're planning to use VCS as the default vivado simulator, make sure you set the VCS_HOME and LM_LICENSE_FILE environment variables before launching Vivado. Note: you can also do it after you've launched Vivado by setting those two variables in the TCL console in Vivado.

```
export VCS_HOME=<path to vcs-mx/O-2018.09-SP2-6>
export LM_LICENSE_FILE=<your license file>
```

Packaging BlackParrot as a custom IP in Vivado

Add design sources to a project:

1) Check the reference documents below on how to start the basic steps
2) While selecting the default part of your project, you may select **Boards** and select **Genesys2** (Fig. 1) if you're working with Genesys2 boards. Hit **Next** and **Finish**.
3) Under the **Sources** window, you can add all of the BlackParrot related files to generate a BlackParrot IP. Right-click anywhere in the **Sources** window and select **Add Sources..**
4) Select **Add or Create Design Sources -> Add Files** and add all the files you need. Note: Be careful while adding a ton of files as it would take the Vivado tool a long time to Update the Hierarchy and Compile Order
5) Vivado tool automatically detects file hierarchy, for example, bp_be has many submodules and Vivado will automatically add those files under bp_be once they're added as design sources.
6) Notice how adding a design file may cause a syntax error in Vivado (Fig. 2). That is because the file is written in SystemVerilog but has a file extension of .v. Vivado thinks that this is a Verilog file and doesn't understand the syntax. To

resolve this issue, go to the **Source File Properties** window and change the file type from Verilog to SystemVerilog.

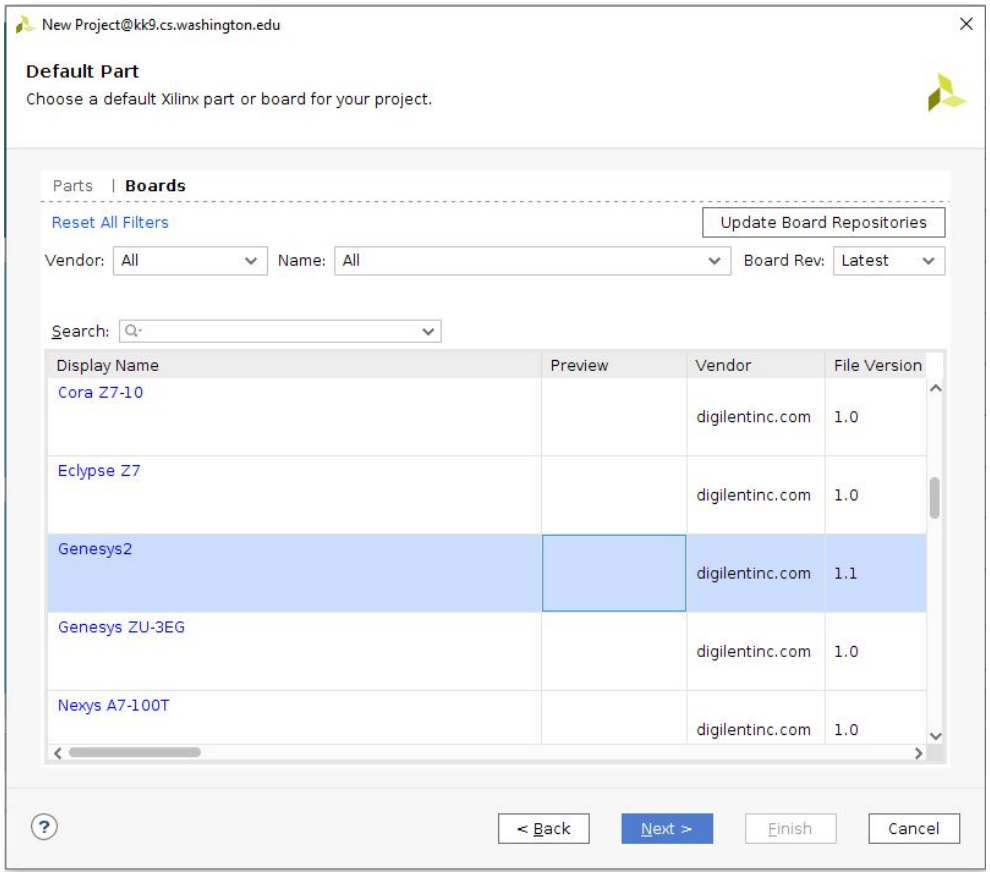

Fig. 1 Selecting Parts/Boards when creating a project

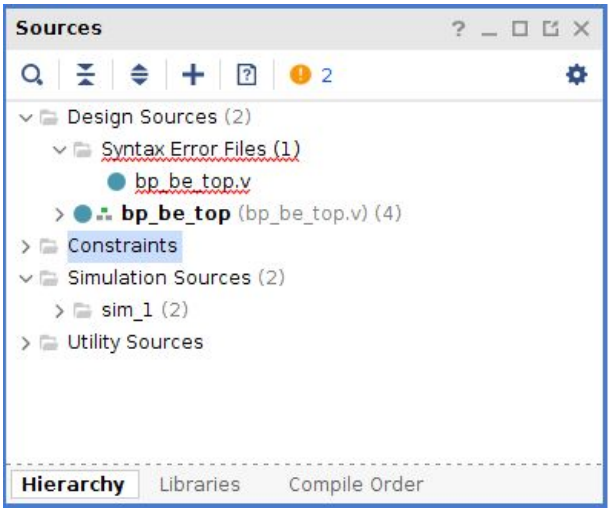

Fig. 2 Syntax error due to incorrect file type

7) You may still see some syntax error but that is ok because we have not yet added the dependent packages.
8) Now go ahead and add all of the files

Creating an IP with design source files:

1) Once you've added all the files, go to **Tools** on the top and select **Create and Package New IP**
2) Choose **Package your current directory** and select the correct directory for your IP to be packaged
3) A new window will open up, on the left hand side, select **Edit Packaged IP**
4) Go through the packaging steps and modify the parameters accordingly
5) For port interface, start by right clicking anywhere in the Ports and Interface window and choose **Add Bus Interface…** This allows you to better pack all the relevant IOs into a nicely packed bus format for the GUI
6) The Interface Definition allows you to select what kind of interface your signal is. Figure 3 shows a clock signal interface definition.
7) Next, map the correct interface's logic ports to IP physical ports (Fig. 4)
8) Once you're done with the other interfaces, make sure you associate those interfaces with a clock if needed. To do so, right click on the interface and select associate to clock.
9) As of now, there isn't a need for memory address mapping. But in the future, if multiple IO devices are supported, then it's necessary to segment the address so that the AXI interconnect IP will know where to direct the traffic
10) Once everything is done, go ahead and package the IP.

Reference documents:
ug1118-vivado-creating-packaging-custom-ip
ug1119-vivado-creating-packaging-ip-tutorial

Adding BlackParrot to Vivado block design

Import the newly created IP into Vivado IP Catalog

1) When you have your new Vivado project open somewhere and you would like to add your new IP, you must first add a new repository where you've created your IP catalog.
2) Click **IP Catalog** in the Flow Navigator on the left side
3) Once the IP catalog is open, right click anywhere within and select **Add Repository…**
4) Navigate to where the IP is located and hit **Select**
5) The IP should now be added and should be visible in the IP list.

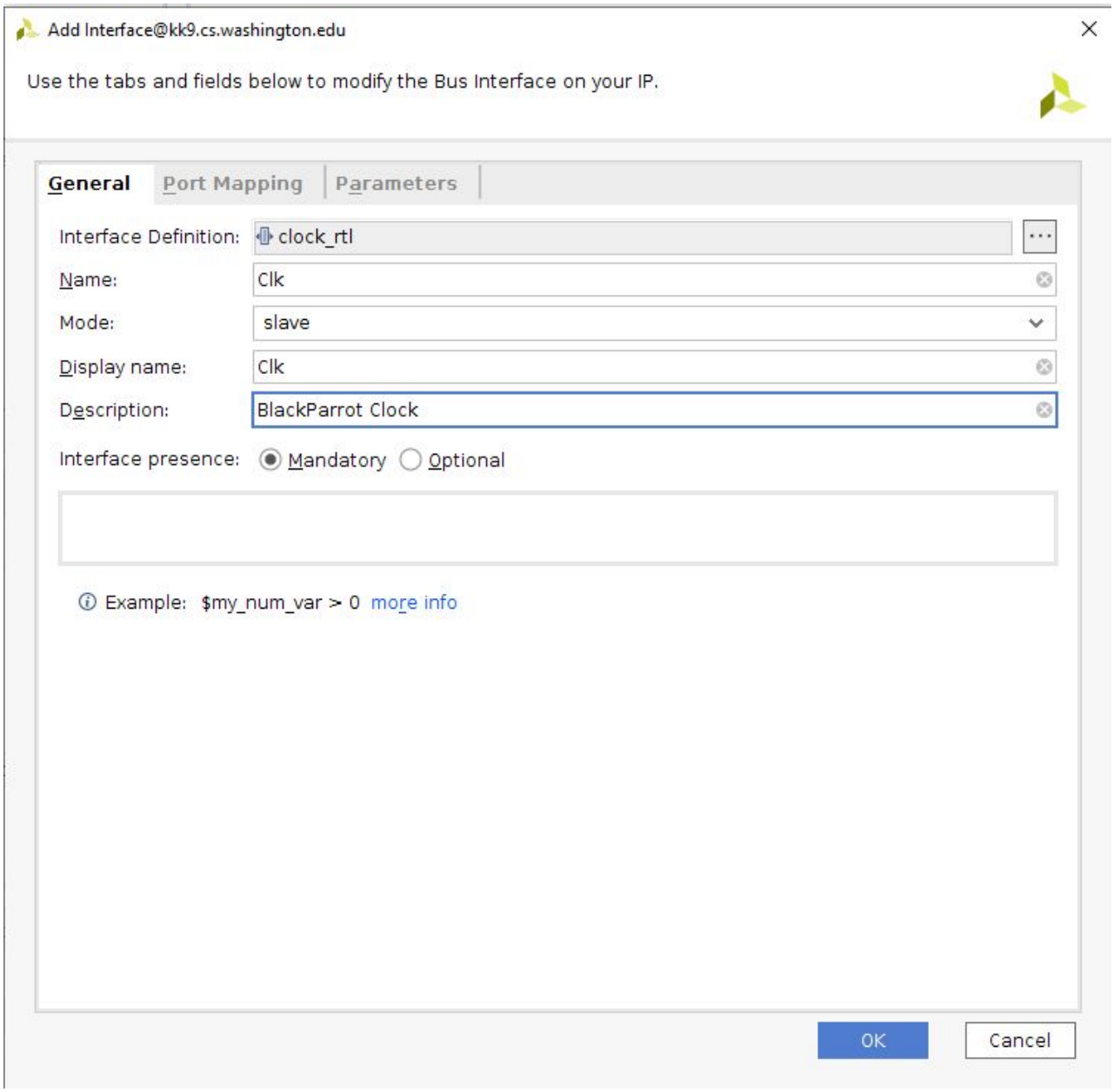

Fig. 3 Clock interface setup

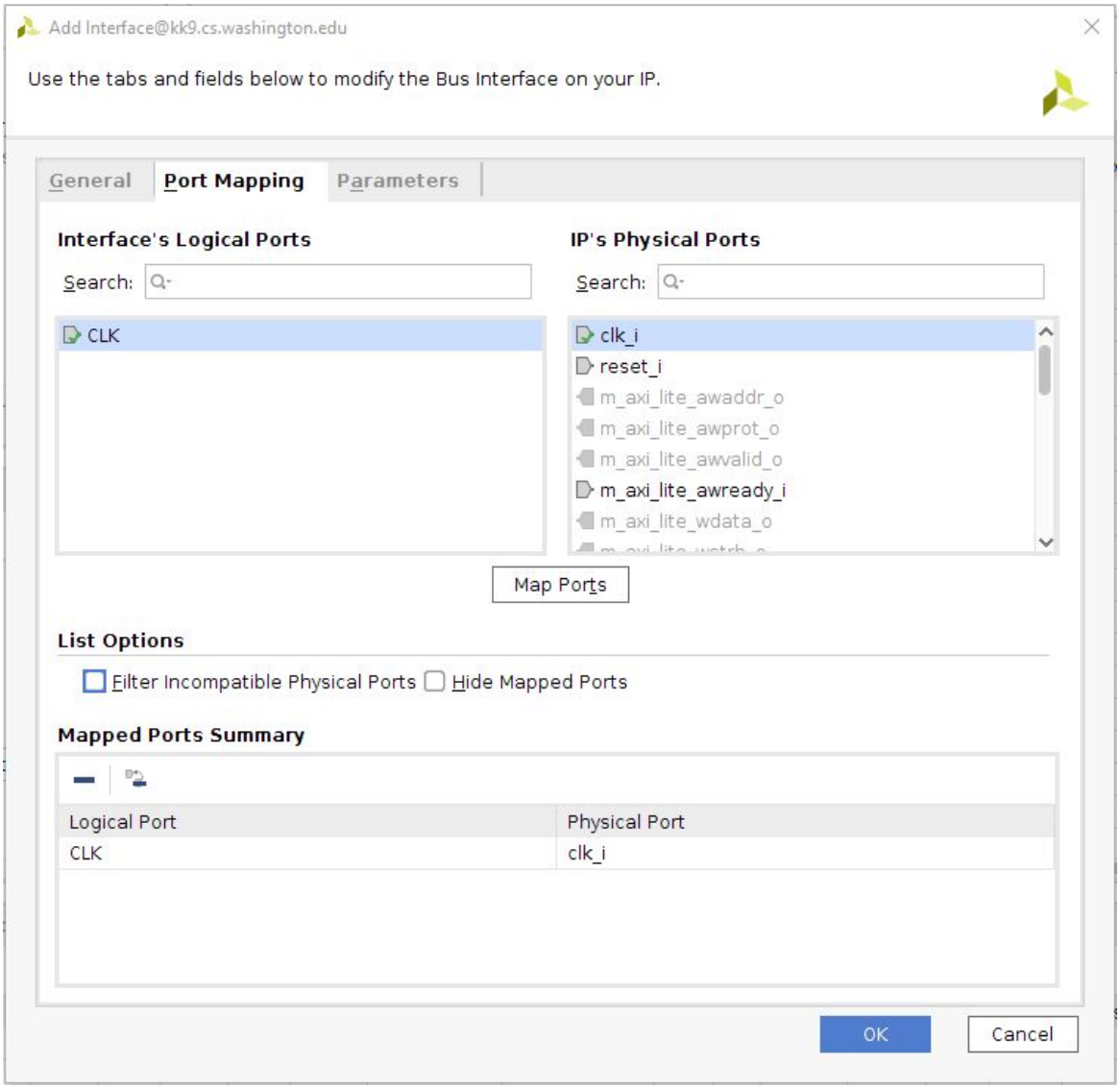

Fig. 4 Clock port mapping

Adding BP as an IP:

1) Create a new block design
2) In the block design window, right click and search for BlackParrot as your design
3) Select the BlackParrot IP and it should now be added to your block design
4) To edit its parameters, double click the BlackParrot IP
5) To edit the files within the IP, right click and select **Edit In Packager**
6) A pop-up window should now appear and you can edit the IP mentioned in the earlier section

Reference documents:

ug994-vivado-ip-subsystems

Block Design Description

Functionality of each IP in the block design:

1) See Fig. 5 - 7for the block design
2) **clk_wiz_100MHz**: use the system differential clock as a source to generate the a 100MHz clock
3) **rst_clk_wiz_100MHz:** generates a 100MHz active high/low synchronous reset
4) **axi_interconnect_0:** an asynchronous FIFO that converts the data from a 100MHz BlackParrot to a 225MHz Memory Interface Generator
5) **Mig_7series_0 (MIG):** a memory interface generator that converts AXI interface signals to DDR3 DRAM controller signals. This IP needs to be manually changed and the values should follow genesys2 board guidelines. Also, check the MIS (Memory Interface Solution) document for more Vivado DDR3 details..
6) **BlackParrot_0:** BlackParrot_V0_2020_12_22 which currently supports all simulation for all unicore configuration
7) For future designs, connect BP IOs to board supported communication IOs (such as UARTLITE or other switches). The UART communication IO can be used to load in programs

Reference documents:
https://reference.digilentinc.com/reference/programmable-logic/genesys-2/reference-manual
ug586_7Seriies_MIS

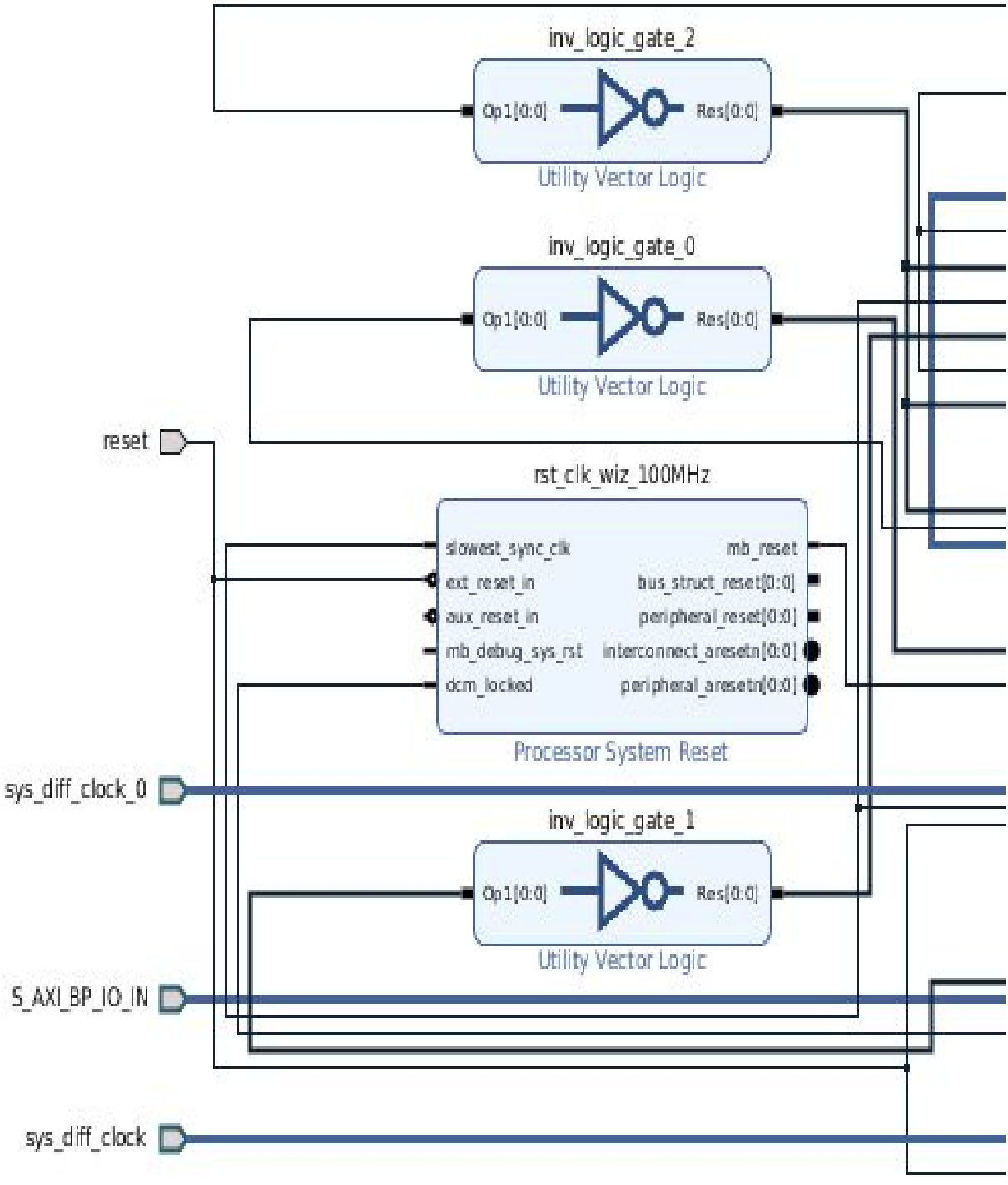

Fig. 5 Block design (left)

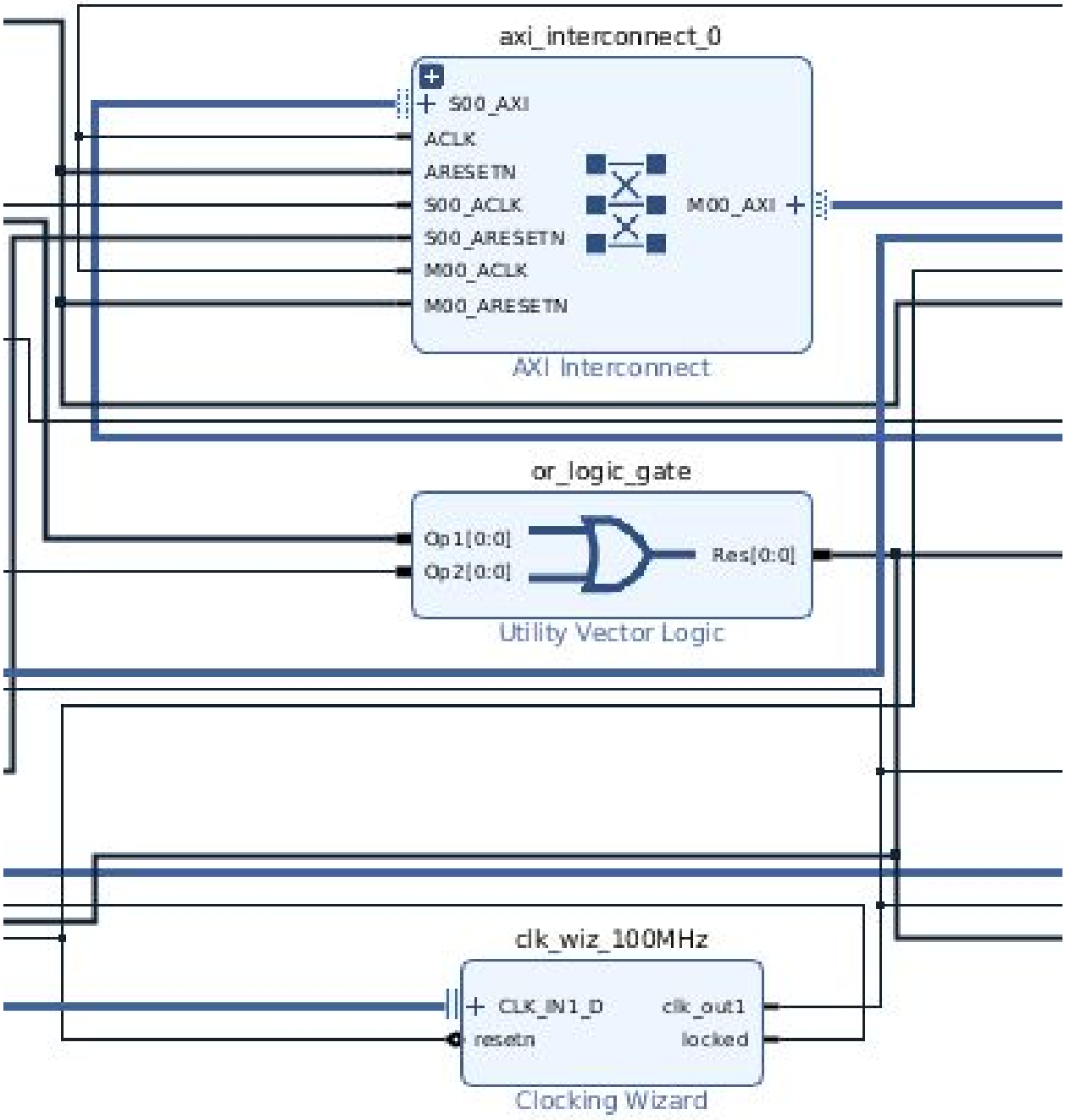

Fig. 6 Block design (middle)

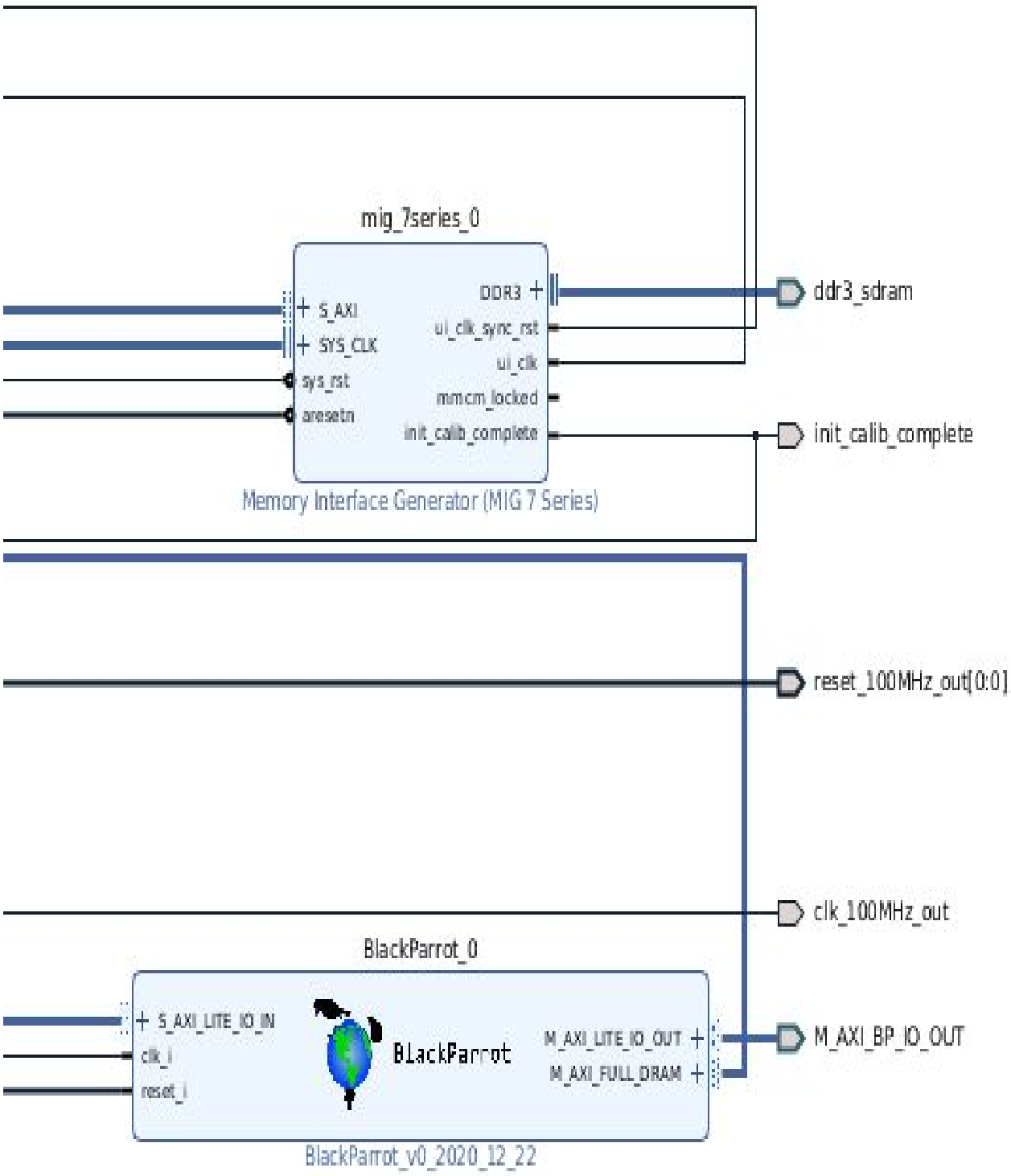

Fig. 7 Block design (right)

Simulation

Simulation model:

1) Check Fig. 8 for simulation testbench setup
2) The simulation consists of a top level testbench that connects BP with a DDR3 model (provided by Xilinx and Micron), a block diagram design (BP + clk/rst + axi interconnect), and a program loader (that is the host + nbf_loader from BP).
3) With the provided scripts, VCS will be chosen as the default simulator with the simulation time set to 1ms. The proj_setup.tcl will set up the compilation, elaboration, and simulation flags
4) To add more VCS flags, click **Tools -> Setting -> Simulation**
5) Fig. 9 whos all the tabs in the simulation settings window
6) After the block diagram is design and all the ip files are generated, we have to add all of the simulation sources located in the bp_packaged_ip/tb
7) Vivado should automatically identify the hierarchy of the files
8) Make sure you set “sim_tb_top.v” as your top

Simulation breakdown:

1) BP waits for the MIG to be calibrated before doing anything. The calibration usually takes about 50,000,000ps to complete. Once the calibration is completed, BP will come out of reset
2) The DRAM’s memory in the region of 0x0 - 0xffff is initialized to 0 through a file called “mem_init.txt”
3) The nbf files are loaded into BP through the bp_nonsynth_tb_top (host + nbf_loader). Prog.nbf and bootrom.mem are located in the bp_packaged_IP/tb
4) Once the simulation is launched, it will go through compilation, elaboration, and finally simulation.

Simulation hints/tips:

1) Simulation may be slow so be patient. Try out the smaller programs first to ensure it is working
2) Many components in the original blackparrot non-synth tb are commented out, like importing C functions. Thus you may not see many of the printouts from black-parrot RTL simulation. Don’t freak out, you should still be able to see PASS/FAIL when the test is done.
3) You may also increase the BP operating frequency to higher than 100MHz, though that has led to other problems that you need to solve such as available FIFO slots in the AXI interconnect IP.

4) Sequences such as 5555, 9999, AAAA, 6666 are written into the DRAM during its initial calibration process. So don't be afraid about seeing those values initially in the simulation printout.

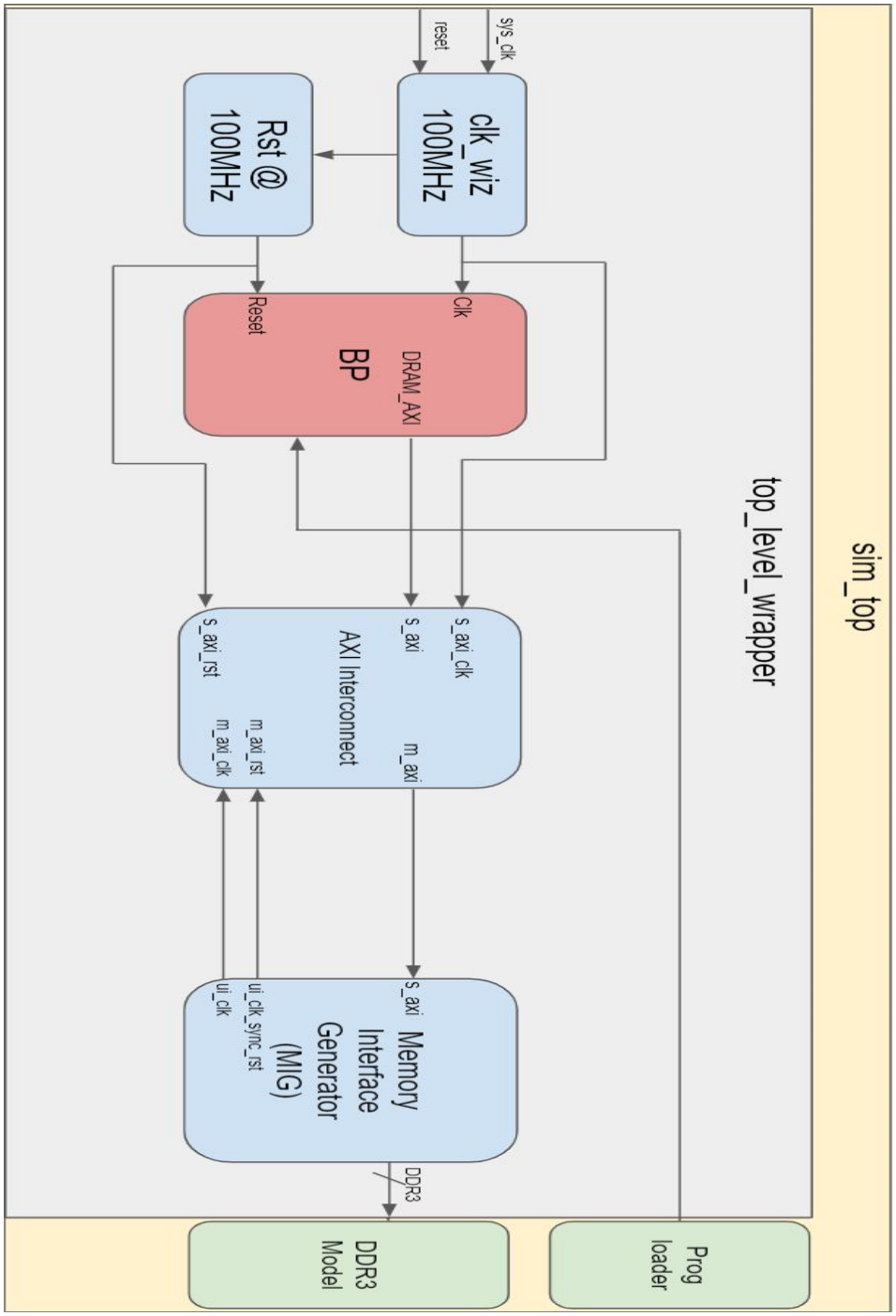

Fig. 8 BlackParrot simulation setup

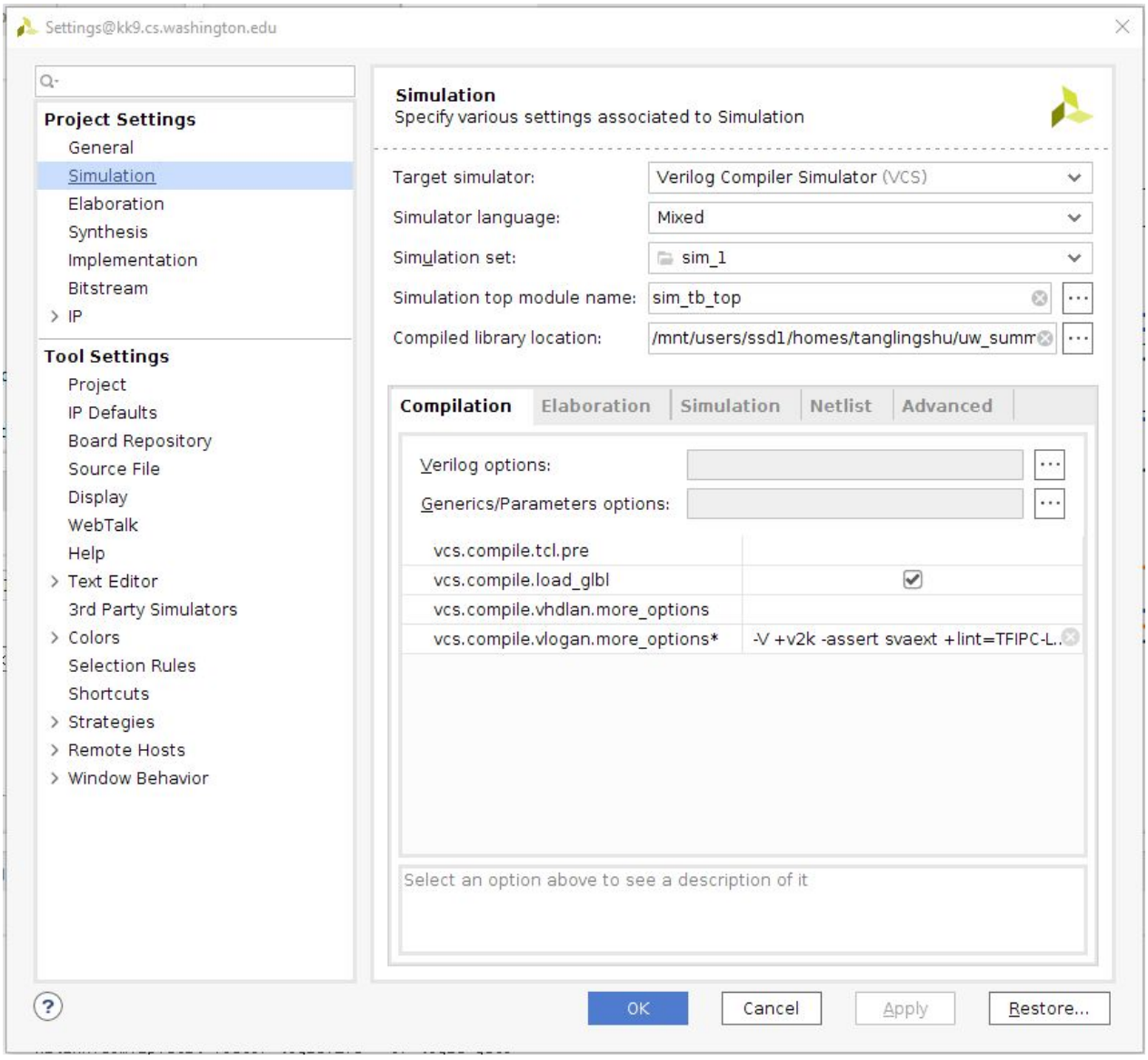

Fig. 9 Simulation settings window

Other reference documents:

IHI0022E_amba_axi_and_ace_protocol_spec

pg267-axi-vip

TCL Script Draft

```
# set your paths and variables
set bp_ip_path  <path_to_your_bp_ip_location>
set vcs_bin_path <path_to_vcs_bin>
set vcs_lib_path <path_to_where_you_want_your_compiled_vcs_lib>
set ::env(VCS_HOME) <path_to_your_vcs_home>
set ::env(LM_LICENSE_FILE) <your_license_key>
set proj_name <proj_name_of_your_choice>
set proj_path <path_to_your_proj>

# compile vcs library to use vcs as a simulator in Vivado
compile_simlib -simulator vcs -simulator_exec_path $vcs_bin_path -family all -language
all -library all -dir $vcs_lib_path -no_ip_compile

# project creation
create_project $proj_name $proj_path -part xc7k325tffg900-2
set_property board_part digilentinc.com:genesys2:part0:1.1 [current_project]

# create block diagram design
create_bd_design "design_1"

# 100 MHz Clock
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:clk_wiz:6.0 clk_wiz_0
set_property name clk_wiz_100MHz [get_bd_cells clk_wiz_0]
set_property -dict [list CONFIG.CLK_IN1_BOARD_INTERFACE {sys_diff_clock}
CONFIG.RESET_BOARD_INTERFACE {reset} CONFIG.RESET_TYPE
{ACTIVE_LOW} CONFIG.PRIM_SOURCE {Differential_clock_capable_pin}
CONFIG.CLKIN1_JITTER_PS {50.0} CONFIG.MMCM_CLKFBOUT_MULT_F {5.000}
CONFIG.MMCM_CLKIN1_PERIOD {5.000} CONFIG.MMCM_CLKIN2_PERIOD {10.0}
CONFIG.RESET_PORT {resetn} CONFIG.CLKOUT1_JITTER {112.316}
CONFIG.CLKOUT1_PHASE_ERROR {89.971}] [get_bd_cells clk_wiz_100MHz]
apply_bd_automation -rule xilinx.com:bd_rule:board -config { Board_Interface
{sys_diff_clock ( System differential clock ) } Manual_Source {Auto}}  [get_bd_intf_pins
clk_wiz_100MHz/CLK_IN1_D]
apply_bd_automation -rule xilinx.com:bd_rule:board -config { Board_Interface {reset (
Reset ) } Manual_Source {New External Port (ACTIVE_LOW)}}  [get_bd_pins
clk_wiz_100MHz/resetn]
endgroup
```

```
# 100 MHz Processor Reset
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:proc_sys_reset:5.0 proc_sys_reset_0
set_property name rst_clk_wiz_100MHz [get_bd_cells proc_sys_reset_0]
set_property -dict [list CONFIG.RESET_BOARD_INTERFACE {reset}] [get_bd_cells
rst_clk_wiz_100MHz]
apply_bd_automation -rule xilinx.com:bd_rule:clkrst -config {Clk
"/clk_wiz_100MHz/clk_out1 (100 MHz)" }  [get_bd_pins
rst_clk_wiz_100MHz/slowest_sync_clk]
apply_bd_automation -rule xilinx.com:bd_rule:board -config { Board_Interface {reset (
Reset ) } Manual_Source {Auto}}  [get_bd_pins rst_clk_wiz_100MHz/ext_reset_in]
connect_bd_net [get_bd_pins clk_wiz_100MHz/locked] [get_bd_pins
rst_clk_wiz_100MHz/dcm_locked]
endgroup

# BlackParrot
set_property  ip_repo_paths  $bp_ip_path [current_project]
update_ip_catalog

startgroup
create_bd_cell -type ip -vlnv bjump.org:user:BlackParrot:0.2020.12.22 BlackParrot_0
set_property -dict [list CONFIG.axi_lite_data_width_p {64}] [get_bd_cells BlackParrot_0]
connect_bd_net [get_bd_pins BlackParrot_0/clk_i] [get_bd_pins
clk_wiz_100MHz/clk_out1]
endgroup

# AXI Interconnect
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:axi_interconnect:2.1 axi_interconnect_0
set_property -dict [list CONFIG.NUM_MI {1} CONFIG.ENABLE_ADVANCED_OPTIONS
{1}] [get_bd_cells axi_interconnect_0]
connect_bd_net [get_bd_pins clk_wiz_100MHz/clk_out1] [get_bd_pins
axi_interconnect_0/S00_ACLK]
connect_bd_intf_net [get_bd_intf_pins BlackParrot_0/M_AXI_FULL_DRAM]
-boundary_type upper [get_bd_intf_pins axi_interconnect_0/S00_AXI]
endgroup

# Memory Interface Generator (MIG)
startgroup
```

```
create_bd_cell -type ip -vlnv xilinx.com:ip:mig_7series:4.2 mig_7series_0
apply_board_connection -board_interface "ddr3_sdram" -ip_intf
"mig_7series_0/mig_ddr_interface" -diagram "design_1"
#set_property -name {CONFIG.XML_INPUT_FILE} -value  {mig_a.prj} -objects
[get_bd_cells mig_7series_0]
#set_property -name {CONFIG.RESET_BOARD_INTERFACE} -value  {Custom}
-objects [get_bd_cells mig_7series_0]
#set_property -name {CONFIG.MIG_DONT_TOUCH_PARAM} -value  {Custom}
-objects [get_bd_cells mig_7series_0]
#set_property -name {CONFIG.BOARD_MIG_PARAM} -value  {ddr3_sdram} -objects
[get_bd_cells mig_7series_0]
connect_bd_intf_net -boundary_type upper [get_bd_intf_pins
axi_interconnect_0/M00_AXI] [get_bd_intf_pins mig_7series_0/S_AXI]
connect_bd_net [get_bd_ports reset] [get_bd_pins mig_7series_0/sys_rst]
connect_bd_net [get_bd_pins mig_7series_0/ui_clk] [get_bd_pins
axi_interconnect_0/ACLK]
connect_bd_net [get_bd_pins mig_7series_0/ui_clk] [get_bd_pins
axi_interconnect_0/M00_ACLK]
connect_bd_net [get_bd_pins mig_7series_0/aresetn] [get_bd_pins
inv_logic_gate_2/Res]
endgroup

# Other logic gates and net connection
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:util_vector_logic:2.0 util_vector_logic_0
set_property name inv_logic_gate_0 [get_bd_cells util_vector_logic_0]
set_property -dict [list CONFIG.C_SIZE {1} CONFIG.C_OPERATION {not}
CONFIG.LOGO_FILE {data/sym_notgate.png}] [get_bd_cells inv_logic_gate_0]
copy_bd_objs /  [get_bd_cells {inv_logic_gate_0}]
copy_bd_objs /  [get_bd_cells {inv_logic_gate_0}]
create_bd_cell -type ip -vlnv xilinx.com:ip:util_vector_logic:2.0 util_vector_logic_0
set_property name or_logic_gate [get_bd_cells util_vector_logic_0]
set_property -dict [list CONFIG.C_SIZE {1} CONFIG.C_OPERATION {or}
CONFIG.LOGO_FILE {data/sym_orgate.png}] [get_bd_cells or_logic_gate]
connect_bd_net [get_bd_pins mig_7series_0/init_calib_complete] [get_bd_pins
inv_logic_gate_0/Op1]
connect_bd_net [get_bd_pins inv_logic_gate_0/Res] [get_bd_pins or_logic_gate/Op1]
connect_bd_net [get_bd_pins rst_clk_wiz_100MHz/mb_reset] [get_bd_pins
or_logic_gate/Op2]
connect_bd_net [get_bd_pins or_logic_gate/Res] [get_bd_pins BlackParrot_0/reset_i]
```

```
connect_bd_net [get_bd_pins inv_logic_gate_1/Op1] [get_bd_pins or_logic_gate/Res]
connect_bd_net [get_bd_pins inv_logic_gate_1/Res] [get_bd_pins
axi_interconnect_0/S00_ARESETN]
connect_bd_net [get_bd_pins mig_7series_0/ui_clk_sync_rst] [get_bd_pins
inv_logic_gate_2/Op1]
connect_bd_net [get_bd_pins inv_logic_gate_2/Res] [get_bd_pins
axi_interconnect_0/ARESETN]
connect_bd_net [get_bd_pins axi_interconnect_0/M00_ARESETN] [get_bd_pins
inv_logic_gate_2/Res]
set_property CONFIG.ASSOCIATED_BUSIF {S_AXI_BP_IO_IN} [get_bd_ports
/clk_100MHz_out]
set_property CONFIG.ASSOCIATED_BUSIF {S_AXI_BP_IO_IN:M_AXI_BP_IO_OUT}
[get_bd_ports /clk_100MHz_out]

endgroup

# IO interface
startgroup
create_bd_intf_port -mode Slave -vlnv xilinx.com:interface:aximm_rtl:1.0
S_AXI_BP_IO_IN
set_property -dict [list CONFIG.CLK_DOMAIN {design_1_clk_wiz_0_0_clk_out1}
CONFIG.PROTOCOL {AXI4LITE} CONFIG.DATA_WIDTH {64} CONFIG.HAS_BURST
{0} CONFIG.HAS_CACHE {0} CONFIG.HAS_LOCK {0} CONFIG.HAS_QOS {0}
CONFIG.HAS_REGION {0}] [get_bd_intf_ports S_AXI_BP_IO_IN]
connect_bd_intf_net [get_bd_intf_ports S_AXI_BP_IO_IN] [get_bd_intf_pins
BlackParrot_0/S_AXI_LITE_IO_IN]
create_bd_intf_port -mode Master -vlnv xilinx.com:interface:aximm_rtl:1.0
M_AXI_BP_IO_OUT
set_property -dict [list CONFIG.CLK_DOMAIN {design_1_clk_wiz_0_0_clk_out1}
CONFIG.PROTOCOL {AXI4LITE} CONFIG.DATA_WIDTH {64} CONFIG.HAS_BURST
{0} CONFIG.HAS_CACHE {0} CONFIG.HAS_LOCK {0} CONFIG.HAS_QOS {0}
CONFIG.HAS_REGION {0}] [get_bd_intf_ports M_AXI_BP_IO_OUT]
connect_bd_intf_net [get_bd_intf_ports M_AXI_BP_IO_OUT] [get_bd_intf_pins
BlackParrot_0/M_AXI_LITE_IO_OUT]
create_bd_port -dir O init_calib_complete
connect_bd_net [get_bd_ports init_calib_complete] [get_bd_pins
mig_7series_0/init_calib_complete]
create_bd_port -dir O -type rst reset_100MHz_out
connect_bd_net [get_bd_ports reset_100MHz_out] [get_bd_pins or_logic_gate/Res]
create_bd_port -dir O -type clk clk_100MHz_out
```

```
connect_bd_net [get_bd_ports clk_100MHz_out] [get_bd_pins
clk_wiz_100MHz/clk_out1]
endgroup

# Excluding memory segment
exclude_bd_addr_seg [get_bd_addr_segs BlackParrot_0/S_AXI_LITE_IO_IN/Reg]
-target_address_space [get_bd_addr_spaces S_AXI_BP_IO_IN]

# Block design cleanup and validation
regenerate_bd_layout
save_bd_design
validate_bd_design

# Create HDL wrapper
startgroup
set dir [get_property DIRECTORY [current_project]]
set name [get_property NAME [current_project]]
set bdFile [get_files ${dir}/${name}.srcs/sources_1/bd/*.bd]
make_wrapper -files [get_files $bdFile] -top
add_files -norecurse [get_files
${dir}/${name}.srcs/sources_1/bd/design_1/hdl/design_1_wrapper.v]
endgroup

# Generate Output Products
export_ip_user_files -of_objects [get_ips design_1_BlackParrot_0_0] -no_script -sync
-force -quiet
generate_target all [get_files $bdFile]
export_simulation -of_objects [get_files
${dir}/project_4.srcs/sources_1/bd/design_1/design_1.bd] -directory
${dir}/project_4.ip_user_files/sim_scripts -ip_user_files_dir ${dir}/project_4.ip_user_files
-ipstatic_source_dir ${dir}/project_4.ip_user_files/ipstatic -lib_map_path [list
{modelsim=${dir}/project_4.cache/compile_simlib/modelsim}
{questa=${dir}/project_4.cache/compile_simlib/questa}
{ies=${dir}/project_4.cache/compile_simlib/ies}
{xcelium=${dir}/project_4.cache/compile_simlib/xcelium} {vcs=${vcs_lib_path}}
{riviera=${dir}/project_4.cache/compile_simlib/riviera}] -use_ip_compiled_libs -force
-quiet

-use_ip_compiled_libs -force -quiet
report_ip_status -name ip_status
```

```
launch_runs -jobs 72 design_1_synth_1

# Setting VCS as default simulator
set_property target_simulator VCS [current_project]
set_property compxlib.vcs_compiled_library_dir $vcs_lib_path [current_project]

# Adding simulation files
set_property SOURCE_SET sources_1 [get_filesets sim_1]
add_files -fileset sim_1 -norecurse [join “
  $bp_ip_path/tb/axi_lite_to_bp_lite_client.sv
  $bp_ip_path/tb/bp_be_pkg.sv
  $bp_ip_path/tb/bp_common_aviary_defines.svh
  $bp_ip_path/tb/bp_common_aviary_pkg.sv
  $bp_ip_path/tb/bp_common_bedrock_if.svh
  $bp_ip_path/tb/bp_common_cache_engine_if.svh
  $bp_ip_path/tb/bp_common_core_if.svh
  $bp_ip_path/tb/bp_common_csr_defines.svh
  $bp_ip_path/tb/bp_common_defines.svh
  $bp_ip_path/tb/bp_common_pkg.sv
  $bp_ip_path/tb/bp_common_rv64_defines.svh
  $bp_ip_path/tb/bp_lite_to_axi_lite_master.sv
  $bp_ip_path/tb/bp_me_pkg.sv
  $bp_ip_path/tb/bp_nonsynth_host.sv
  $bp_ip_path/tb/bp_nonsynth_if_verif.sv
  $bp_ip_path/tb/bp_nonsynth_nbf_loader.sv
  $bp_ip_path/tb/bp_nonsynth_tb_top.sv
  $bp_ip_path/tb/bp_nonsynth_tb_top_wrapper.v
  $bp_ip_path/tb/bsg_buf.sv
  $bp_ip_path/tb/bsg_bus_pack.sv
  $bp_ip_path/tb/bsg_circular_ptr.sv
  $bp_ip_path/tb/bsg_clkgate_optional.sv
  $bp_ip_path/tb/bsg_counter_up_down.sv
  $bp_ip_path/tb/bsg_decode.sv
  $bp_ip_path/tb/bsg_decode_with_v.sv
  $bp_ip_path/tb/bsg_defines.v
  $bp_ip_path/tb/bsg_dff_en.sv
  $bp_ip_path/tb/bsg_dff_reset_en.sv
  $bp_ip_path/tb/bsg_dff_reset.sv
  $bp_ip_path/tb/bsg_dff.sv
  $bp_ip_path/tb/bsg_dlatch.sv
```

```
  $bp_ip_path/tb/bsg_fifo_1r1w_small_hardened.sv
  $bp_ip_path/tb/bsg_fifo_1r1w_small.sv
  $bp_ip_path/tb/bsg_fifo_1r1w_small_unhardened.sv
  $bp_ip_path/tb/bsg_fifo_tracker.sv
  $bp_ip_path/tb/bsg_flow_counter.sv
  $bp_ip_path/tb/bsg_mem_1r1w.sv
  $bp_ip_path/tb/bsg_mem_1r1w_sync.sv
  $bp_ip_path/tb/bsg_mem_1r1w_sync_synth.sv
  $bp_ip_path/tb/bsg_mem_1r1w_synth.sv
  $bp_ip_path/tb/bsg_muxi2_gatestack.sv
  $bp_ip_path/tb/bsg_mux.sv
  $bp_ip_path/tb/bsg_nand.sv
  $bp_ip_path/tb/bsg_noc_links.vh
  $bp_ip_path/tb/bsg_noc_pkg.v
  $bp_ip_path/tb/bsg_nonsynth_test_rom.sv
  $bp_ip_path/tb/bsg_nor3.sv
  $bp_ip_path/tb/bsg_reduce.sv
  $bp_ip_path/tb/bsg_rotate_right.sv
  $bp_ip_path/tb/bsg_strobe.sv
  $bp_ip_path/tb/bsg_two_fifo.sv
  $bp_ip_path/tb/bsg_xnor.sv
  $bp_ip_path/tb/ddr3_model_parameters.vh
  $bp_ip_path/tb/ddr3_model.sv
  $bp_ip_path/tb/sim_tb_top.v
  $bp_ip_path/tb/wiredly.v
  $bp_ip_path/tb/bootrom.mem
  $bp_ip_path/tb/mem_init.txt
  $bp_ip_path/tb/prog.nbf
“]
set_property file_type SystemVerilog [get_files  $bp_ip_path/tb/bsg_noc_pkg.v]
set_property file_type SystemVerilog [get_files  $bp_ip_path/tb/sim_tb_top.v]
set_property file_type {Verilog Header} [get_files  $bp_ip_path/tb/bsg_defines.v]
update_compile_order -fileset sim_1

# Setting VCS simulation flags
set_property -name {vcs.compile.vlogan.more_options} -value {-V +v2k -sverilog -assert
svaext +lint=TFIPC-L} -objects [get_filesets sim_1]
set_property -name {vcs.elaborate.vcs.more_options} -value {-sverilog -assert svaext
-lca +lint=TFIPC-L +noportcoerce} -objects [get_filesets sim_1]
set_property -name {vcs.simulate.runtime} -value {1ms} -objects [get_filesets sim_1]
```

```
set_property -name {vcs.simulate.log_all_signals} -value {true} -objects [get_filesets
sim_1]

# Launch simulation
launch_simulation -install_path $vcs_bin_path
```