Introduction to Zynq Hardware

Lab 5

Adding a PL Peripheral

November 2013
Version 03

Lab 5 Overview

In this lab, we will expand our block design by extending our memory space with a PL-based Block RAM (BRAM). The BRAM can be used to buffer data going between the PS and PL.

Lab 5 Objectives

When you have completed Lab 5, you will know how to do the following:

- Add a BRAM from the IP Catalog
- Connect AXI peripherals to the Zynq PS.

Experiment 1: Add the PL BRAM from the IP Catalog

This experiment shows how to add the PL BRAM IP from the IP Catalog.

Experiment 1 General Instruction:

Open Vivado Project and Block Design. Add Block Memory to the block design.

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. **Open** the project: **C:\Speedway\ZynqHW\2013_3\ZynqDesign\ZynqDesign.xpr**

3. **Open** the Block Design, *Z_system.bd*

4. With the block design open, click anywhere in the white space of the diagram and right-click then select **Add IP…** Alternatively select the Add IP button on the vertical shortcut bar.

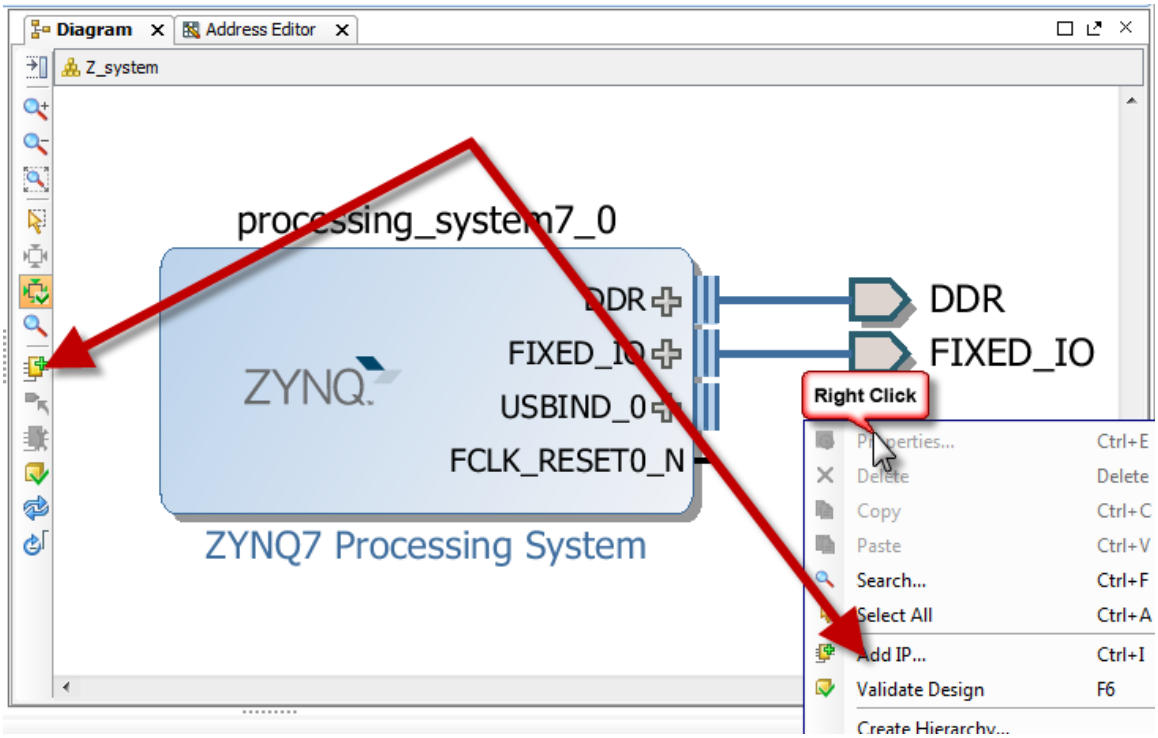


Figure 1 - Add IP to Block Design

5. The IP catalog will open. Add a Block RAM Controller by entering **bram** in the search field. Double-click **AXI BRAM Controller** to add it to the system.

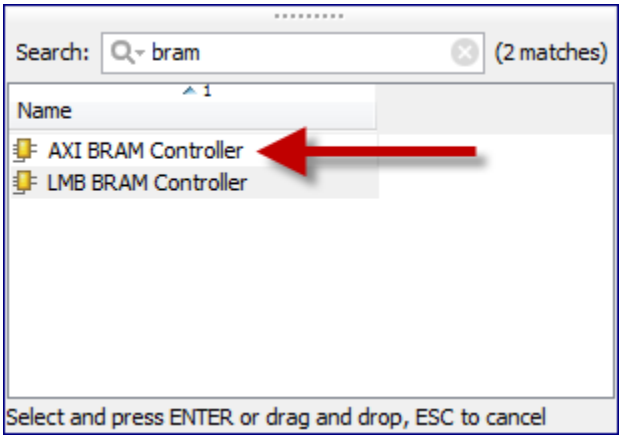


Figure 2 - Add AXI BRAM Controller

6. Double-click the **AXI BRAM Controller**. It may be helpful to hit the Regenerate Layout Button on the vertical shortcut bar, , to see this view.

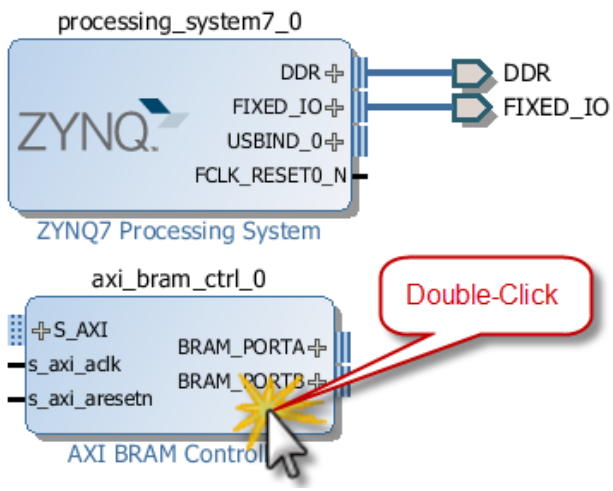


Figure 3 - Customize IP with Double-Click

7. Increase the *Data Width* to **64** bits. Click **OK**.

8. Verify the *ID Width (Auto)* is set to **Yes** for *Support for AXI Narrow Bursts*. If not, switch it to **Manual** and change *Support for AXI Narrow Bursts* to **Yes**. Click **OK**.

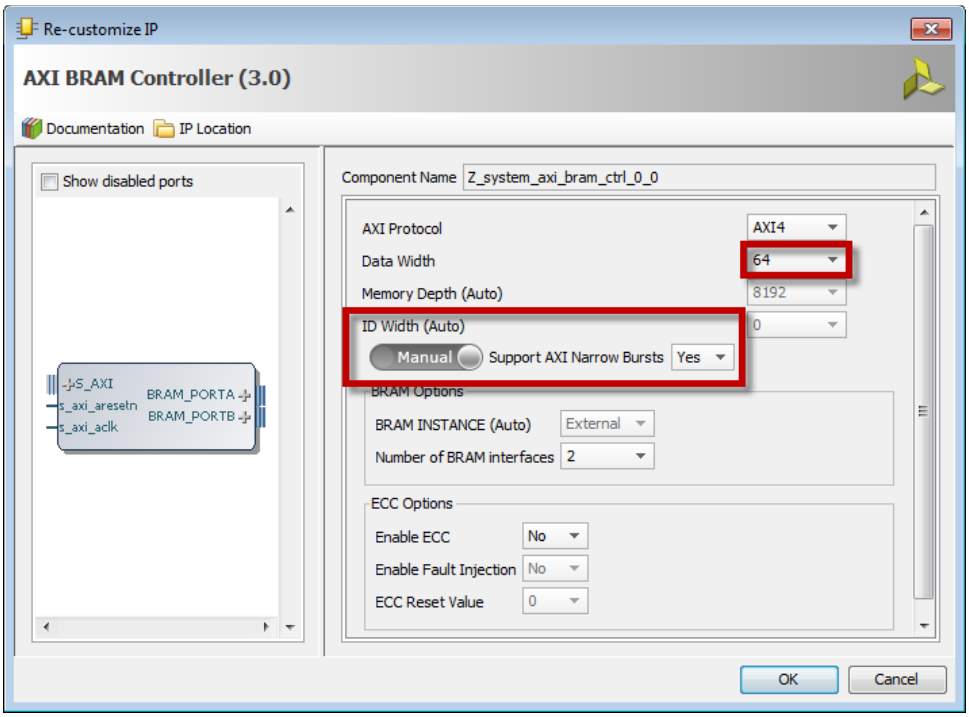


Figure 4 - BRAM Controller Options

9. Next, click Add IP… again and enter **block** in the search field and double-click **Block Memory Generator**.

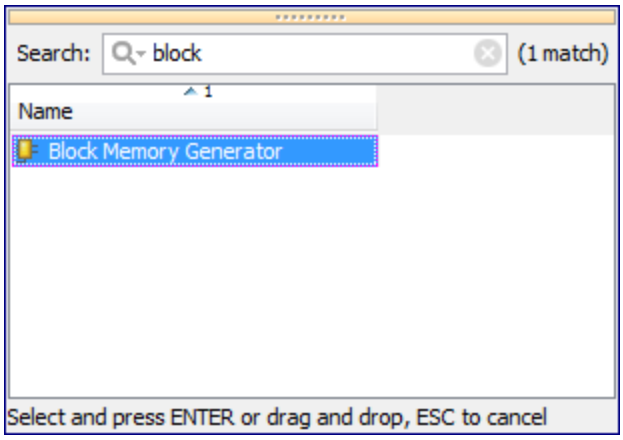


Figure 5 - Add Block Memory Generator

10. Double-click the **blk_mem_gen_0 IP** to customize the IP.

11. Select **True Dual Port** from the *Memory Type* drop-down list. Click **OK**.

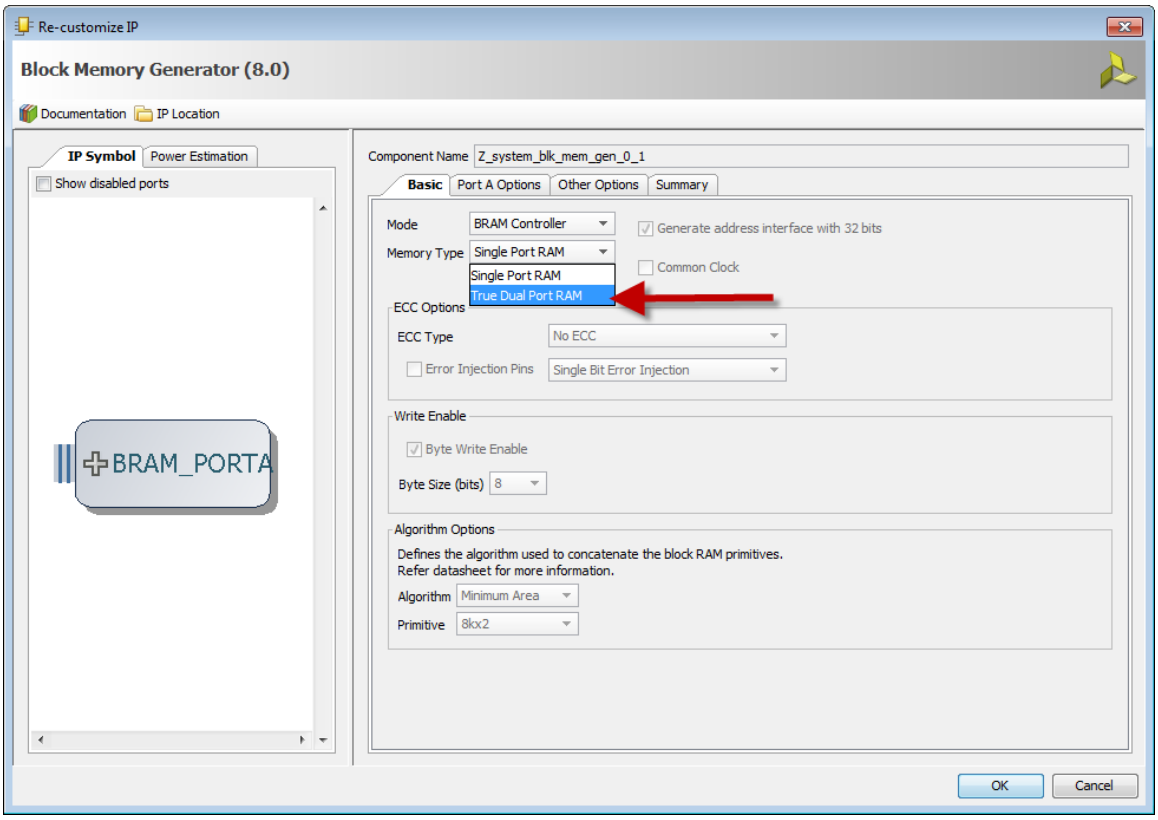


Figure 6 - Configure Block Memory Generator

12. **Connect the two BRAM** ports together by hovering over one of the ports, left-clicking, then dragging to the other port. Notice, when you start drawing nets, green checkboxes will appear over acceptable port connections.

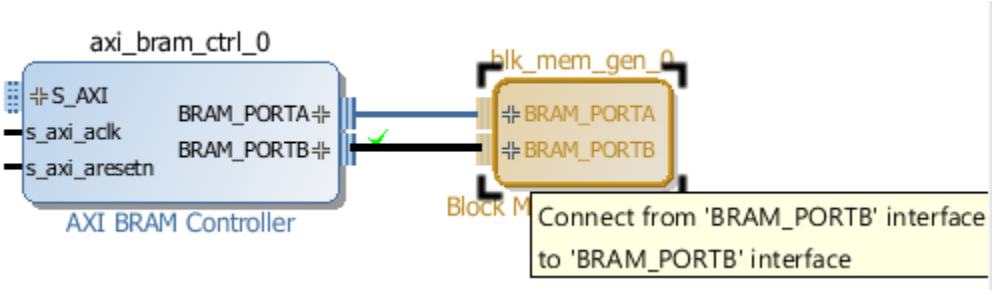


Figure 7 - Connect BRAM Controller to BRAM Generator

Experiment 2: Add AXI Interconnect Block and complete design

This experiment shows how to add an AXI Interconnect Block from the IP Catalog.

Experiment 2 General Instruction:

Add an AXI Interconnect Block and connect all IP to the PS.

Experiment 2 Step-by-Step Instructions:

1. We now need to connect our BRAM to the Zynq PS. To do so, we must add another piece of IP, the AXI Interconnect Block. Click **Add IP**, search for **AXI Int,** then select **AXI Interconnect**.

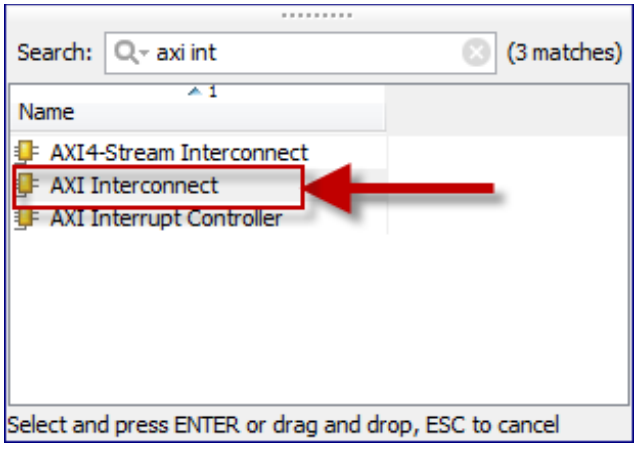


Figure 8 - Add AXI Interconnect IP

2. Customize the AXI Interconnect and change the number of Master interfaces to **one**. Click **OK**.

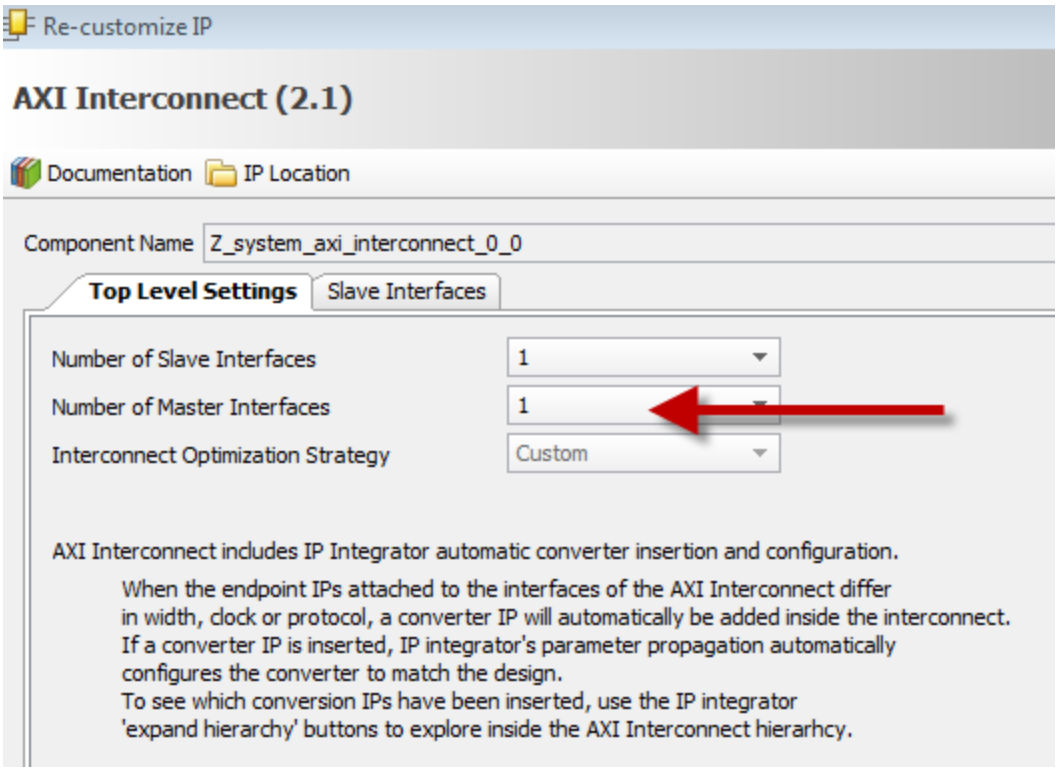


Figure 9 - Customize AXI Interconnect

3. **Connect** the **M00_AXI port** of the AXI Interconnect to the **S_AXI port** of the AXI BRAM Controller. It may be helpful to hit the Regenerate Layout Button on the vertical shortcut bar, .

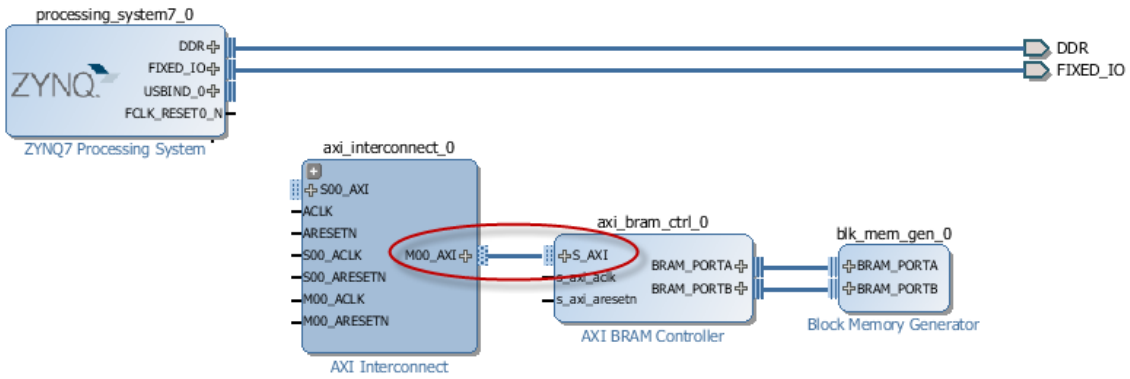


Figure 10 - Connect AXI Interconnect to AXI BRAM

We have many unconnected ports remaining. Most of these are clocks and resets. But the Zynq PS does not have external AXI port, nor does it have an output clock. We must customize the PS to add these ports.

4. **Double-click** the Zynq PS.

5. At the bottom of the screen click on green box labeled, **32b GP AXI Master Ports**.

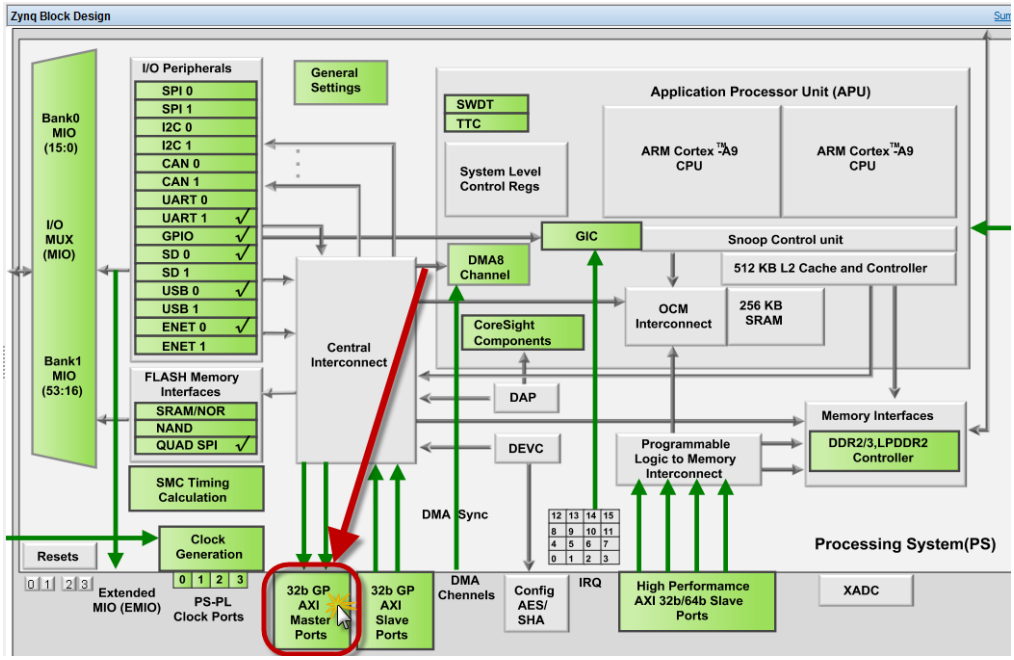


Figure 11 - Customize PS

6. Check the box to select **M AXI GP0 Interface.**

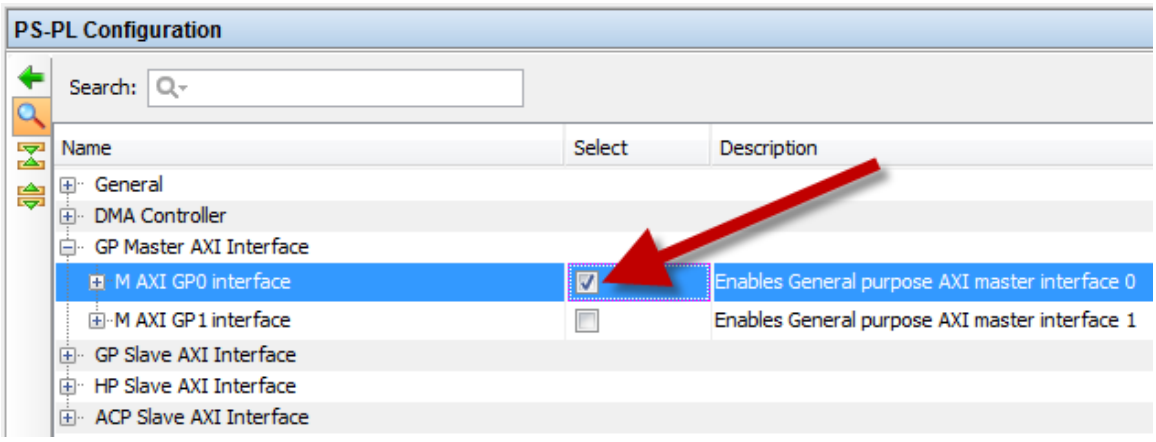


Figure 12 - Enable General Purpose Master AXI Interface

That takes care of our AXI interface that we were missing. Next we need to export a clock from the PS.

7. Select **Clock Configuration** from the Page Navigator. And **expand all** clocks.

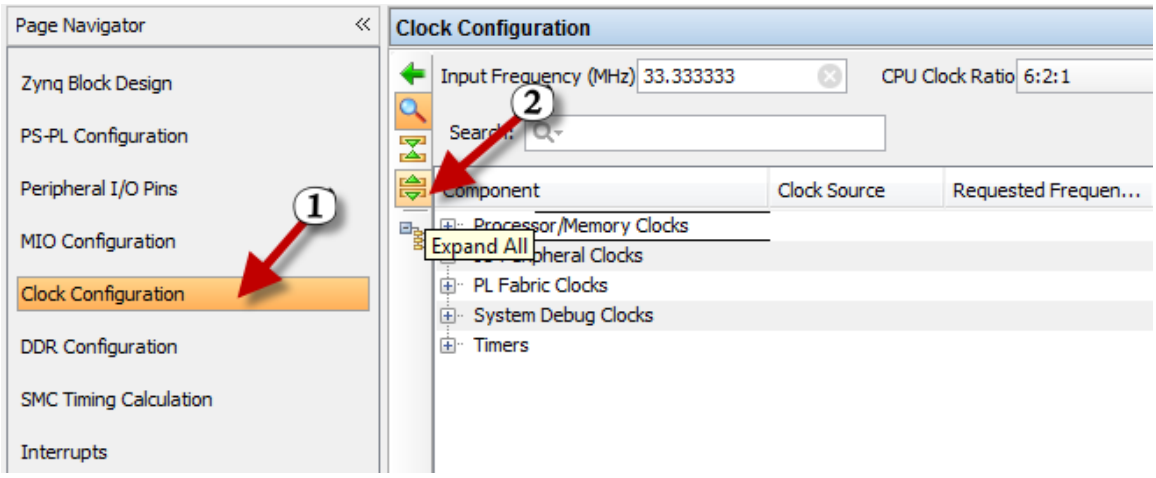


Figure 13 - Clock Configuration

8. Enable Fabric Clock 0 by checking the box next to *FCLK_CLK0*. Set it to **50MHz**.

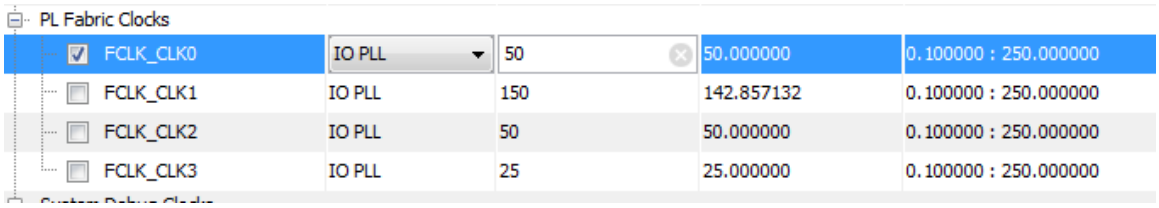


Figure 14 - Enable Fabric Clock 0

9. Click **OK** to exit the PS IP configuration.

10. Some new ports exist now on our PS block. Connect **FCLK_RESET0_N** from the PS to all reset ports on other IP blocks. You should make four connections:

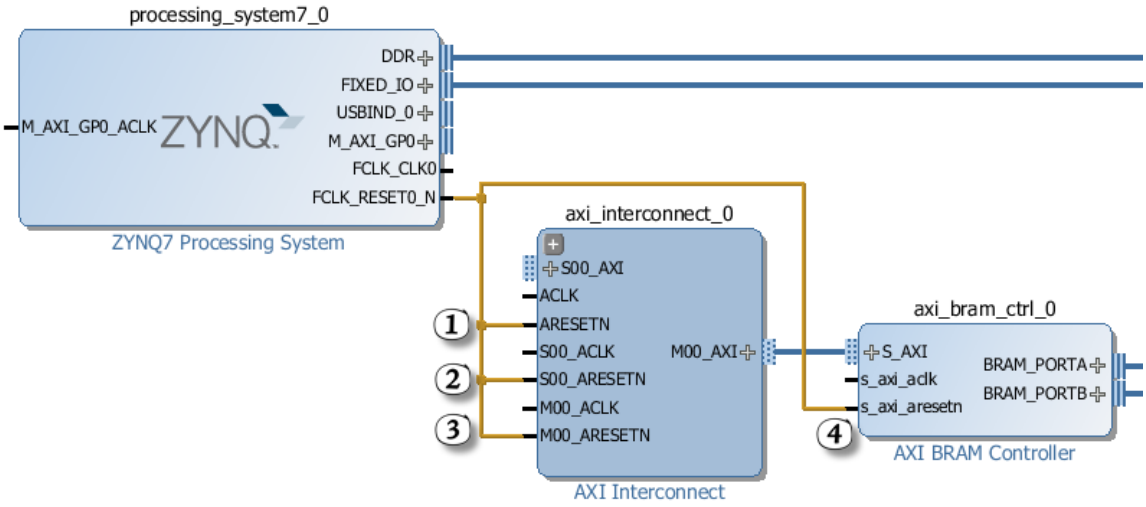


Figure 15 - Connect Resets

11. Connect **FCLK_CLK0** to all clock ports. You should make five connections:

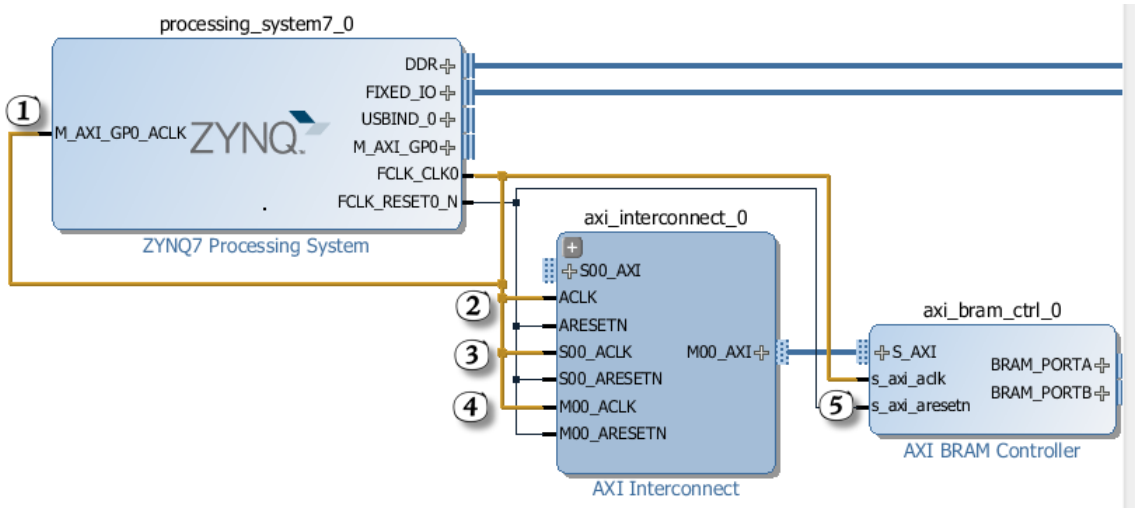


Figure 16 - Connect Clocks

12. The only thing remaining is the AXI connection. Connect **M_AXI_GP0** from the PS to the **S00_AXI** of AXI Interconnect:

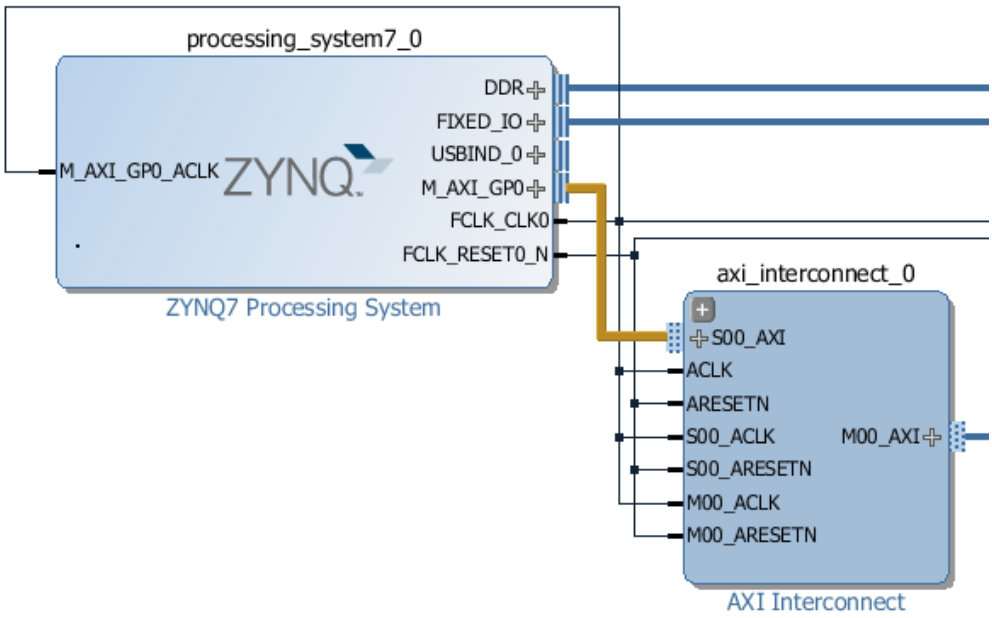


Figure 17 - AXI Connection between PS and AXI Interconnect Block

13. Select **Regenerate Layout** from the vertical shortcut bar. The fully connected design is shown here.

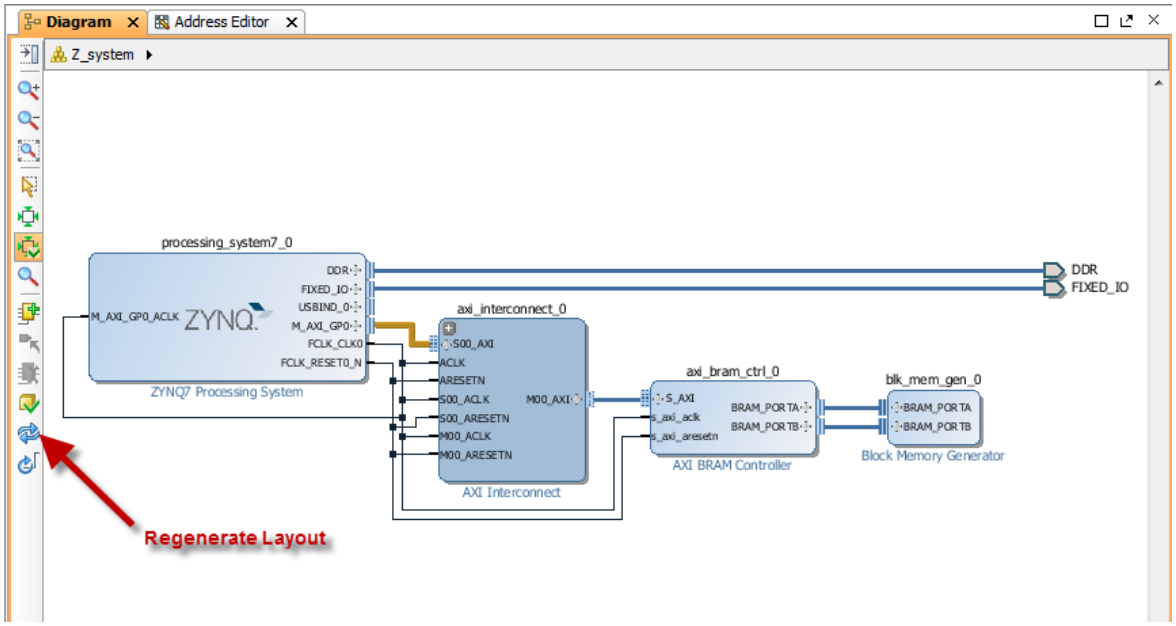


Figure 18 - Fully connected block design

14. We're not quite done yet. We need to create address space for the BRAM on the AXI interface. To do this, select the **Address Editor** tab, **expand all**, then **Auto Assign Address** from the vertical shortcut bar.

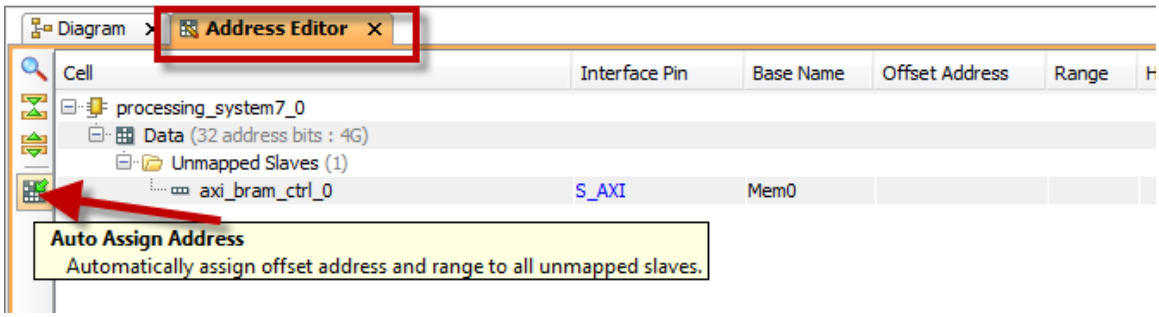


Figure 19 - Assign Address to BRAM

15. Switch back to the *Diagram view* by selecting the **Diagram** tab.

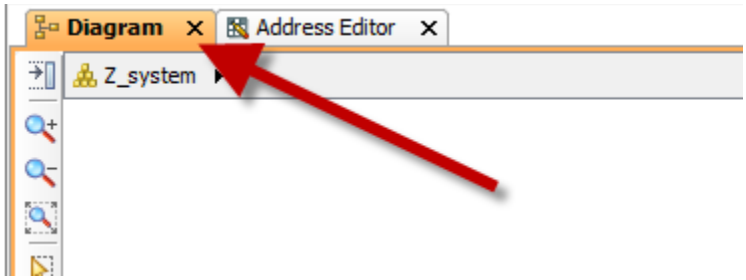


Figure 20 - Diagram Tab

16. Next, we need to validate the design. Vivado has a robust design rule checker for Vivado IP Integrator block designs. Select **Validate Design**. Click **OK** on Validation Successful window.

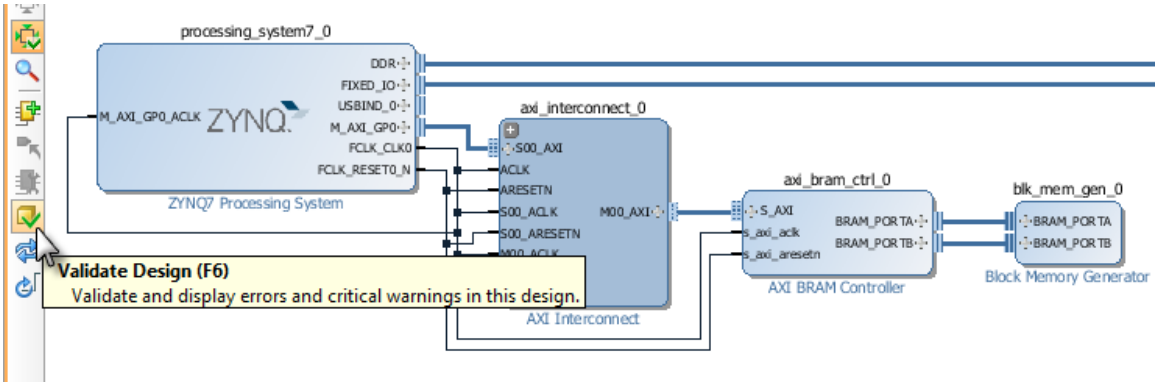


Figure 21 - Validate Design

17. If you see this critical warning, Click **OK** to accept it. This is a bug that does not affect our design. Every time this warning appears just accept it by clicking OK. It will appear several times though synthesis and implementation.

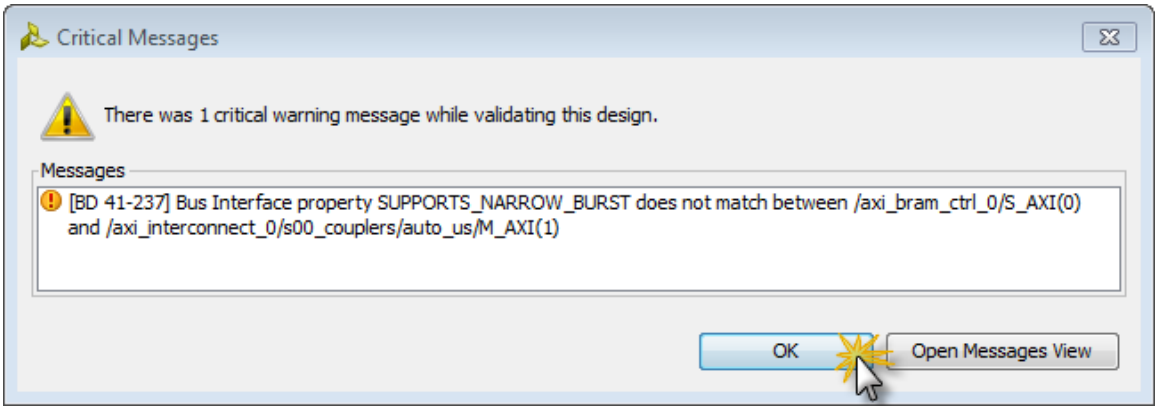
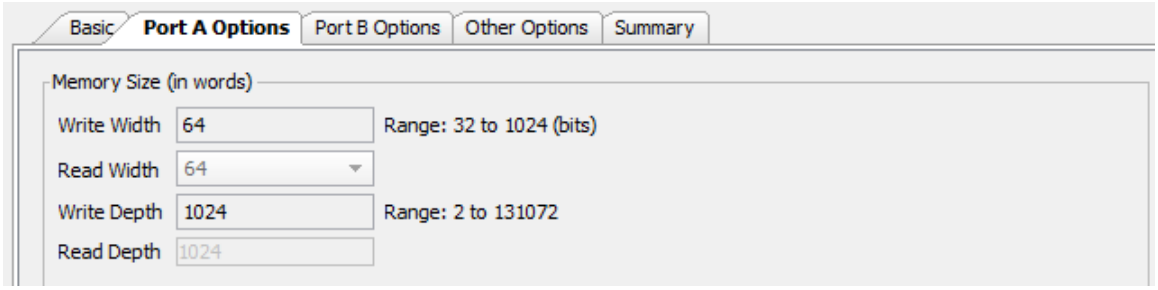
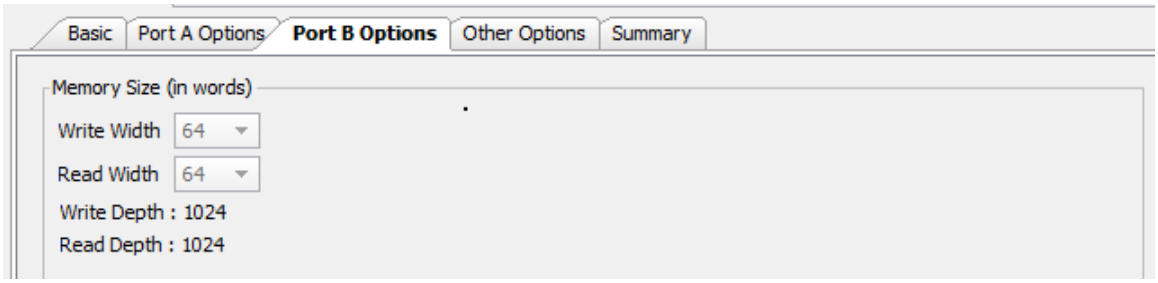


Figure 22 - Ignore Warning

18. Re-customize the Block Memory Generator IP, **blk_mem_gen_0**, by double-clicking on the IP.

19. Click on the **Port A and B Option** tabs to see the data widths and depth. Also click on the **Summary** tab to see how many BRAM resources will be used by this IP. **Note**: The width and depth parameters are inherited from the BRAM Controller IP and updated when the design was validated.





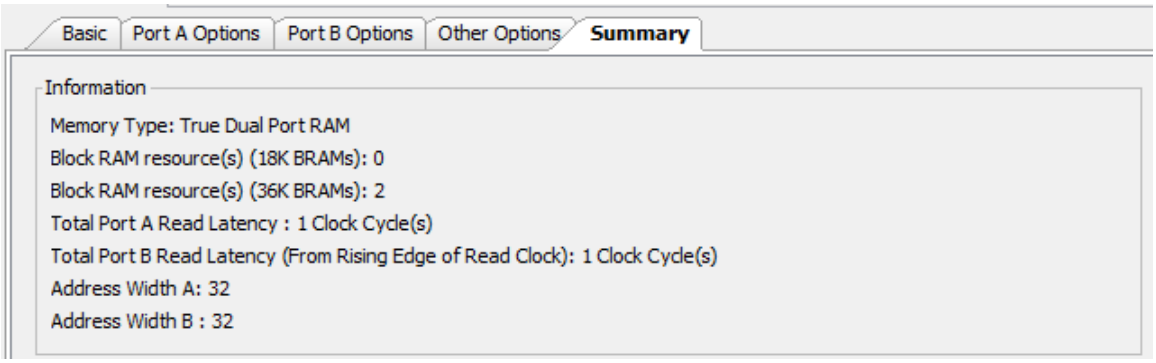


Figure 23 -BRAM Options

20. Click **OK** to accept the settings.

21. Save the block design by typing **save_bd_design** in the TCL Console or by clicking the save block design button from the top shortcut bar.

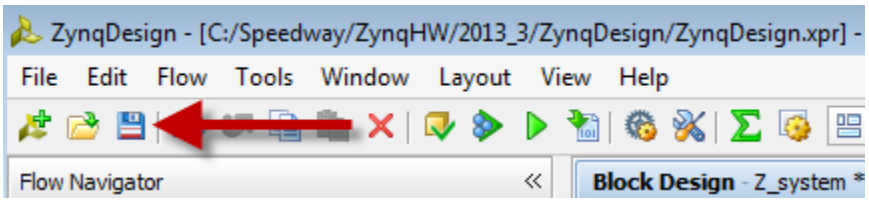


Figure 24 - Save Block Design

22. Select **Generate Bitstream** from the Flow Navigator. Click **Yes** to launch synthesis and implementation. This will take a few minutes. Answer the questions below while waiting.

Questions:

Answer the following questions:

- *How many BRAM's are consumed by the Block Memory Generator?*

_____

- *What is the base address of the BRAM? Why is it mapped here?*

_____

- *If more IP peripherals were connected, where would they connect?*

_____

Exploring Further

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

- Explore what other IP is available in the catalog. Note: if you add it to your design, make sure to remove it before proceeding to the next lab.

This concludes Lab 5.

Revision History

Date	Version	Revision
6 Nov 13	02	Initial Draft
19 Nov 13	03	Pilot updates

Resources

www.microzed.org

www.zedboard.org

www.xilinx.com/zynq

www.xilinx.com/sdk

www.xilinx.com/vivado

Answers

Experiment 2

- *How many BRAM's are consumed by the Block Memory Generator?*

2

- *What is the base address of the BRAM? Why is it mapped here?*

0x40000000, because this is starting address space for M_AXI_GP0 (refer to the Zynq All Programmable SoC User Guide).

- *If more IP peripherals were connected, where would they connect?*

Add Master AXI ports to the AXI Interconnect Block for additional slaves to connect into.