Commas and Data Alignment Lab

Introduction

In this lab, you will use commas to control data flow and align serial data into bytes.

Objectives

After completing this lab, you will be able to:

- Define a data alignment comma
- Send and receive commas by using the RocketIO™ GTP transceiver
- Align a multi-byte datapath

Procedure

In this lab, you will define a data alignment comma and use it for alignment and flow control. You will also align a 2-byte datapath.

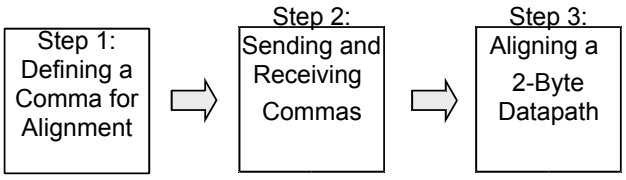
This lab is separated into steps, followed by general instructions and supplementary detailed steps allowing you to make choices based on your skill level as you progress through the lab.

If you need help completing a general instruction, go to the detailed steps below it, or if you are ready, simply skip the step-by-step directions and move on to the next general instruction.

This lab comprises three primary steps: You will define a comma for alignment, send and receive commas, and align a 2-byte datapath.

Note: If you are unable to complete the lab at this time, you can download the original lab files for this module from the Xilinx FTP site atftp://ftp.xilinx.com/pub/documentation/education/rio22000-11-rev1-xlnx_lab_files.zip. These are the original lab files and do not contain any work you may have previously completed.

General Flow for this Lab



1Defining a Comma for Alignment Step 1

1.1. Launch the Project Navigator in the ISE® software and open the *gtp_comma* project in the ..*labs**04_so_lab_comma*\ *directory.*

1.1.1. Select **Start** > **All Programs** > **Xilinx ISE Design Suite** > **ISE** > **Project Navigator**.

1.1.2. Select **File** > **Open Project** in the Project Navigator.

- **VHDL users targeting the ISim simulator:** Browse to .. *labs**04_so_lab_comma*\ directory..

1.1.3. Browse to the *C:\training\rio\labs\Lab2_comma*\<*simulator*>\<*language*> directory and select ***gtp_comma.xise***.

1.1.4. Click **Open**.

1.1.5. Examine the configuration of the GTP component *gtp_inst*.

1.1.6. In the Hierarchy window, select the **Sources for: Implementation** option

1.1.7. Double-click ***lab2.xco***.

The RocketIO GTP Wizard launches and displays the current configuration of this GTP component.

The GTP Wizard will be discussed in more detail later in this course.

Notice that both transceivers are set to use a 2-byte datapath.

1.1.8. Click **Next** until the RX Comma Alignment dialog box displays (page 5 of 11).

Question 1

What is the current alignment comma? Use the Appendix in the user guide (*C:\training\rio\documents\ug386.pdf*) to translate the binary value into an 8B/10B code word.

Question 2

What is the significance of the Any Byte Boundary setting?

1.1.9. Click **Cancel** to close the GTP Wizard.

1.1.10. Edit the HDL code to change the alignment comma for GTP0 to K28.1.

1.1.11. In the Hierarchy window, select the **Sources for: Simulation** option.

1.1.12. In the Hierarchy window, expand **LAB2_TB** > **gtp_example_top** > **gtp_inst**.

1.1.13. Double-click ***gtp_inst_tile.v/vhd***.

1.1.14. Search for the comma detection attributes for GTP0 and change them to the values listed below.

- MCOMMA_10B_VALUE_0 = **0110000011**
- PCOMMA_10B_VALUE_0 = **1001111100**

1.1.15. Search for the comma detection mask attributes for GTP0 and change this to the values listed below.

- COMMA_10B_ENABLE_0 = **1111111111**

1.1.16. Save and close ***gtp_inst_tile.v/vhd***.

2Sending and Receiving Commas Step 2

2.1.

Run a Xilinx ISim simulation by using the *wave_isim.wcfg* file in the main project directory. Run the simulation for 5 us.

2.1.1. In the Hierarchy window, select the **Sources for: Simulation** option and **Behavioral** from the drop-down list.

2.1.2. Select ***LAB2_TB.v/vhd***.

2.1.3. In the Processes window, expand the **ISim Simulator** process and double-click **Simulate Behavioral Model**.

2.1.4. In the Wave window, right-click and select **Delete**.

All signal names will be deleted.

2.1.5. In ISim, select **File** > **Open**. Select ***wave_isim.wcfg*** and click **Open**. In the pop-up dialog box, select the **Connect to existing** option.

This prepared waveform configuration file loads all the needed signal names.

2.1.6. In ISim, click the **Restart** button to restart the simulation.

2.1.7. In ISim, enter ***5.00us*** and click the **Run For** button. Examine the simulation results (Figure 1).

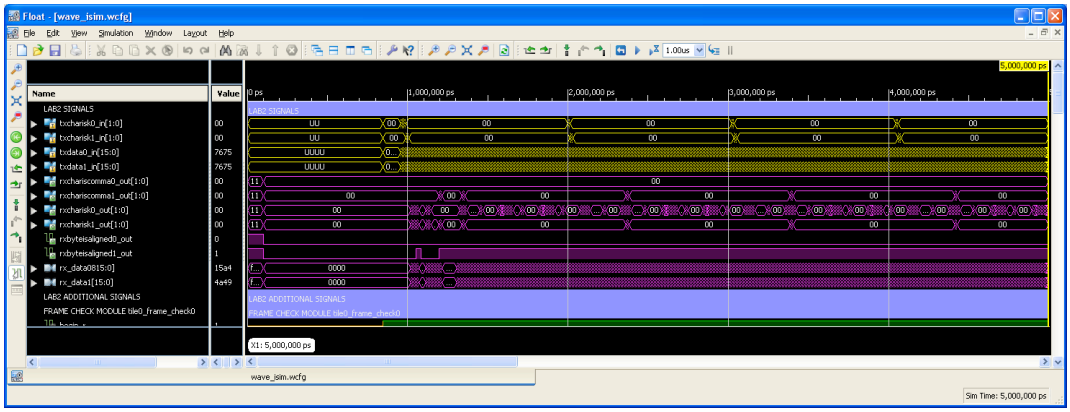


Figure 1. Simulation Results

2.1.8. Close the ISim simulator.

Question 3

Which signals indicate that a data alignment comma has been detected? Which signals indicate that data alignment has occurred?

Question 4

Why is GTP0 not detecting any commas? How can this problem be fixed?

3Edit the block RAM contents in the top-level design file to replace the old comma value with the new one for the GTP0 frame generator.

You do not need to edit the frame checker data because you are not looking at those signals in the simulation.

3.1.1. In the Hierarchy window, double-click ***gtp_example_top.v/vhd***.

3.1.2. Search for the instantiation of the ***tile0_frame_gen0 : FRAME_GEN*** block RAM component.

3.1.3. In the list of *MEM* initialization attributes, replace the *"bc"* at the far right end of every eighth line with *"3c"* (4 characters from the right).

The parity bits are used to drive the ***TXCHARISK*** signal. These attributes do not need to be changed because you are still sending a K character as the data alignment comma.

3.1.4. Save and close ***gtp_example_top.v/vhd***.

3.1.5. Rerun the simulation and observe the comma detection and alignment for both GTP0 and GTP1.

3.1.6. In the Sources window, make sure that you are looking at behavioral simulation sources and select ***Lab2_TB.v/vhd***.

3.1.7. In the Processes window, expand the **ISim Simulator** process and double-click **Simulate Behavioral Model**.

3.1.8. Load the prepared waveform configuration file again, reset the simulation, and run for 5 ns.

3.1.9. Examine the comma detection and data alignment signals and confirm that both transceivers are detecting commas and aligning the data.

3.1.10. Close the ISim simulator.

Question 5

Locate the alignment commas in the RXDATA streams. How is the data alignment different between the two transceivers?

Close the ModelSim simulator.

4Aligning a 2-Byte Datapath Step 3

You will now align the commas.

Recall that in the GTP Wizard, the comma could be aligned to either byte in the data word. Change the ALIGN_COMMA_WORD attribute in the *gtp_inst_tile.v/vhd* file to force comma placement on an even-byte boundary.

4.1.1. In the Hierarchy window, expand ***gtp_example_top*** > ***gtp_inst***.

4.1.2. Double-click ***gtp_inst_tile.v/vhd***.

4.1.3. Search for the two ALIGN_COMMA_WORD attributes and change them both to ***2***.

4.1.4. Save and close ***gtp_inst_tile.v/vhd***.

4.1.5. Restart and run the simulation one final time. Notice how the data alignment for GTP1 has changed.

4.1.6. In the Sources window, make sure that you are looking at behavioral simulation sources and select ***LAB2_TB.v/vhd***.

4.1.7. In the Processes window, expand the **ISim Simulator** process and double-click **Simulate Behavioral Model**.

4.1.8. Examine the *RXDATA** signals and notice that the word alignment for GTP1 has changed (Figure 2 and Figure 3).

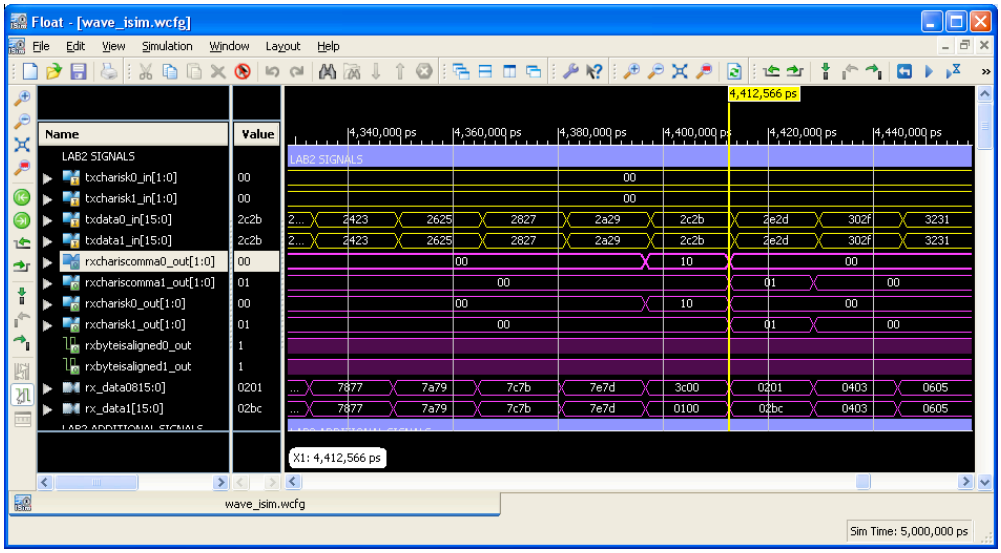


Figure 2. Original Data Alignment

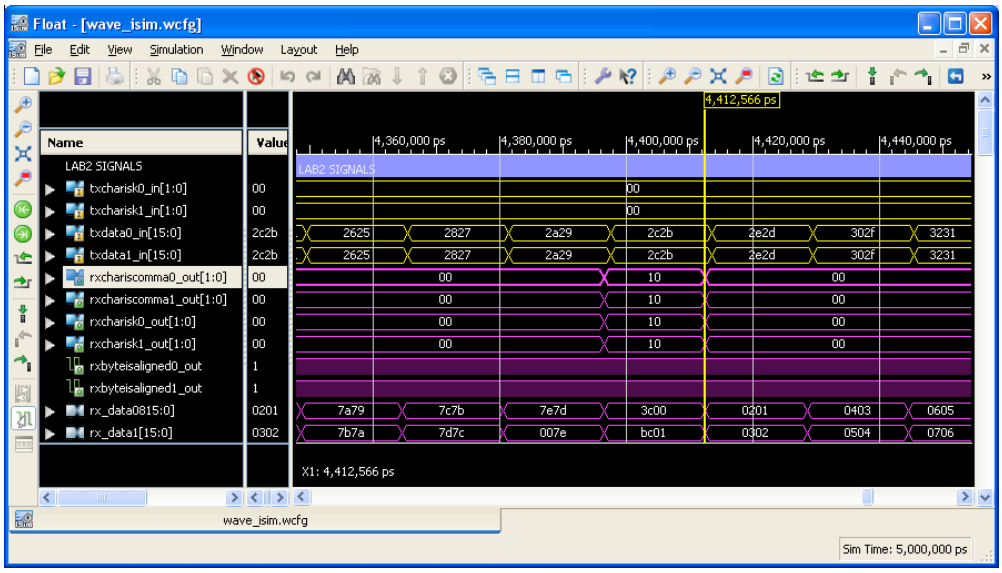


Figure 3. Final Data Alignment

Conclusion

In this lab, you learned how to use commas to align the serial data stream on the proper byte boundaries for 1-byte and 2-byte datapaths.

Answers

1. What is the current alignment comma? Use the Appendix in the user guide (*C:\training\rio\documents\ug386.pdf*) to translate the binary value into an 8B/10B code word.

   The new Spartan®-6 FPGA GTP Wizard gives you the value. Alternatively, use the Appendix in the user guide (*C:\training\rio\documents\ug386.pdf*) to translate the binary value into an 8B/10B code word.

   The plus comma is defined as 0101111100, and the minus comma is defined as 1010000011. Reversing the bit order and using the Appendix in the user guide, this translates to K28.5.

2. What is the significance of the Any Byte Boundary setting?

   This setting indicates that the data alignment comma can occur in either byte of the data word. You will see later in the lab how this can affect alignment and how to set this attribute to align a 2-byte datapath.

3. Which signals indicate that a data alignment comma has been detected? Which signals indicate that data alignment has occurred?

   The ***RXCOMMADET**** signals indicate that an alignment comma has been detected. These signals will be asserted every time an alignment comma is detected.

   The ***RXBYTEISALIGNED**** signals indicate that data alignment has occurred.

   Remember that the ***RXCHARISCOMMA**** signals do not have any relationship to data alignment. These signals simply indicate that one of the pre-defined 8B/10B commas has been detected.

4. Why is GTP0 not detecting any commas? How can this problem be fixed?

   The GTP Wizard inserted comma characters into the block RAM data, but the comma value for GTP0 was then changed in the HDL code. The problem is easily solved by changing the data in the block RAMs to match the current alignment comma.

5. Locate the alignment commas in the RXDATA streams. How is the data alignment different between the two transceivers?

   The data for GTP0 contains an alignment comma in the lower byte of the word, while GTP1 contains a comma in the upper byte of the word. This is allowed because the transceiver was configured to accept an alignment comma in any byte of the data word.

   If you change the transceiver to only allow a comma on an even-byte boundary, the comma received by GTP1 will cause ***RXDATA1*** to be shifted by one byte (or half of a word) relative to ***RXDATA0***.