**ECEN 248 - Lab Report**

**Lab Number: 6**

**Lab Title: Introduction to Behavioral Verilog and Logic Synthesis**

**Section Number: 519**

**Student’s Name: Alex Allahar**

**Student’s UIN: 9280098686**

**Date: 10/25/2023**

**TA: Yi Deng**

**Objectives:**

The purpose of this lab is to introduce a higher level of abstraction in Verilog, called behavioral modeling. Recreating the same multiplexer from the previous lab; however, design then using behavioral modeling. Finally, this lab will introduce the use of programming the ZYBO Z7-10 board, to test comments.

**Design:**

To start, describe a 2:1 mux using behavioral modeling in Verilog. After creating a new project called lab6, create a new design source file called two\_one\_mux\_behavioral and use the code in the background section of the lab as the code for this module. Add the corresponding test bench file from the course directory into the simulation sources of the project, and simulate the module. Once this module is functional create a new module to describe a 4-bit 2:1 mux. In this source start with the code in the previous 2:1 mux, but replace lines 5-11 with the lines provided in the lab manual. Add the corresponding test bench file into the simulation sources of the project from the course directory, and simulate the 4-bit 2:1 mux. Create a new design source file mux\_4bit\_4to1, and use the starter code provided in the lab manual to describe the mux with cases. Copy the test bench file, from the course directory with the same name, and simulate. Verify that the module runs by checking the waveform.

For the next part of the lab, we will build on this by describing a 2:4 binary decoder and a 4:2 binary encoder. Starting with the 2:4 binary decoder, create a module with this name and use the starter code provided to describe the module. Copy the test bench file, from the course directory with the same name, and simulate. Now, describe the 4:2 binary encode starting with the code provided in the lab manual. Copy the test bench file, from the course directory with the same name, and simulate. Create a new design source called the priority encoder, using the code provided in the lab manual as a starting point. Copy the test bench file, from the course directory with the same name, and simulate.

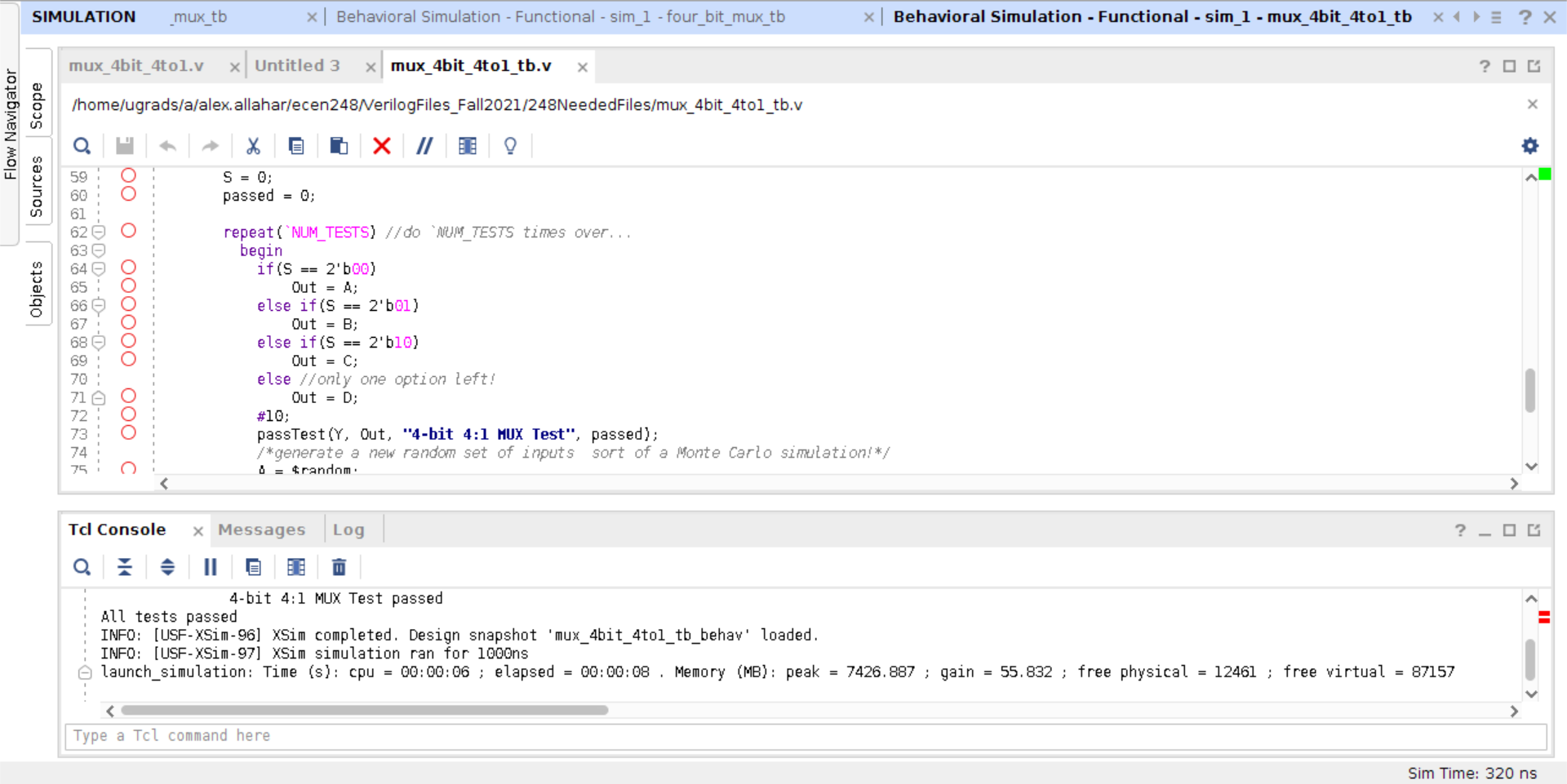
For the last part of the lab, the module will be synthesized and implemented onto the ZYBO Z7-10 board. Copy the constraint file for the 2:4 decoder from the course directory and add it to the project. Set the 2:4 decoder as the top module and generate bitstream.

**Conclusion:**

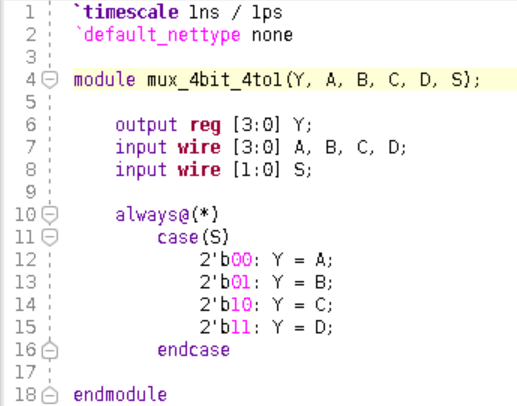
In this lab, I was able to describe circuits with behavioral modeling and was able to translate the Verilog code to a physical output using the ZYBO Z7-10 board.

**Post-lab Deliverables:**

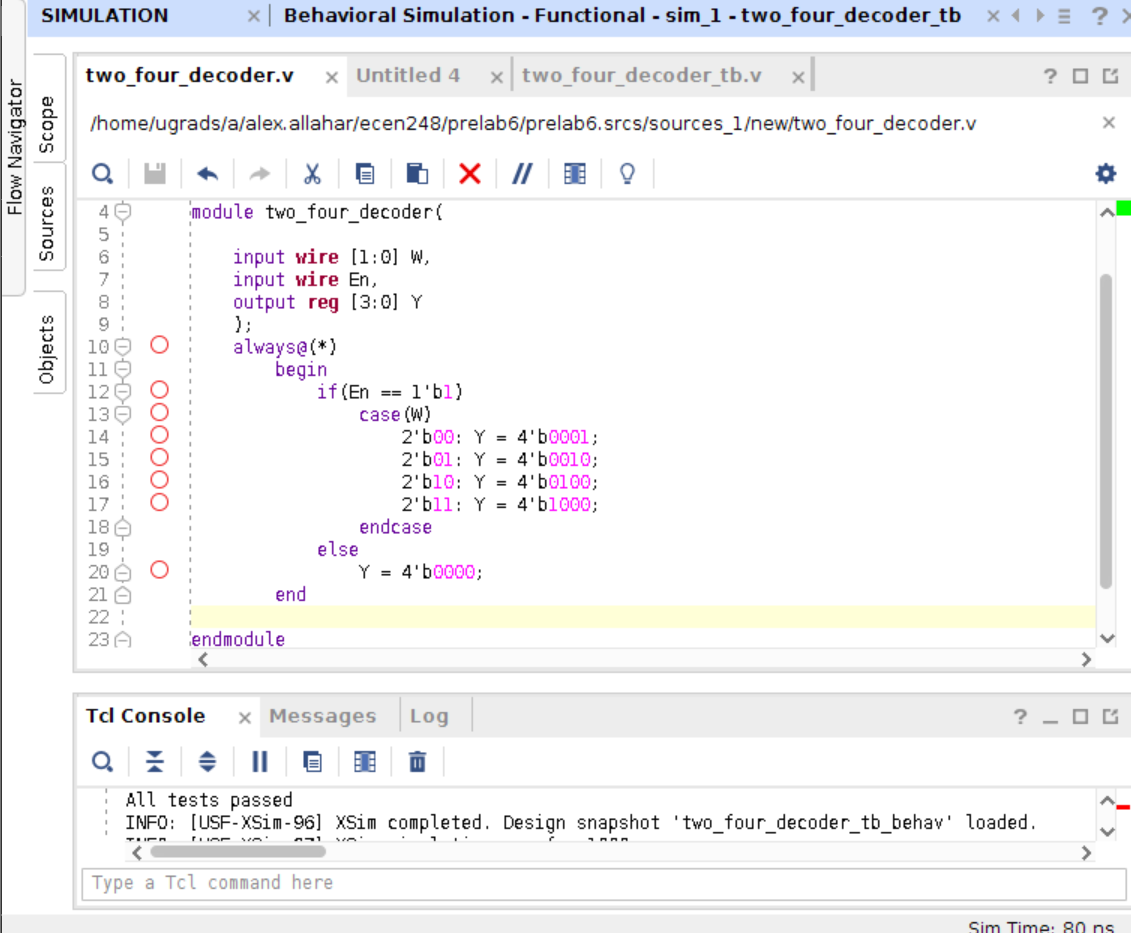
1. **Source Code**

****

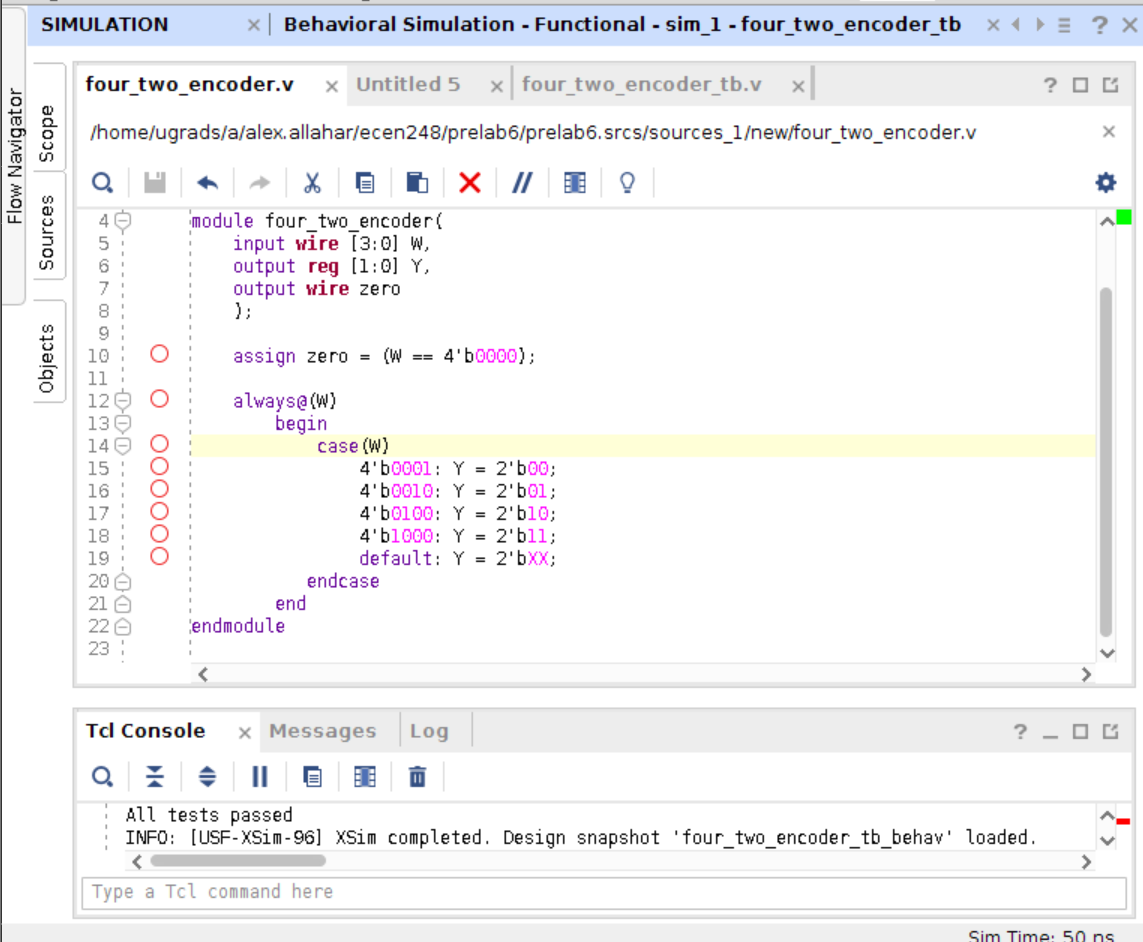
**Figure 1: 4-bit mux code**

****

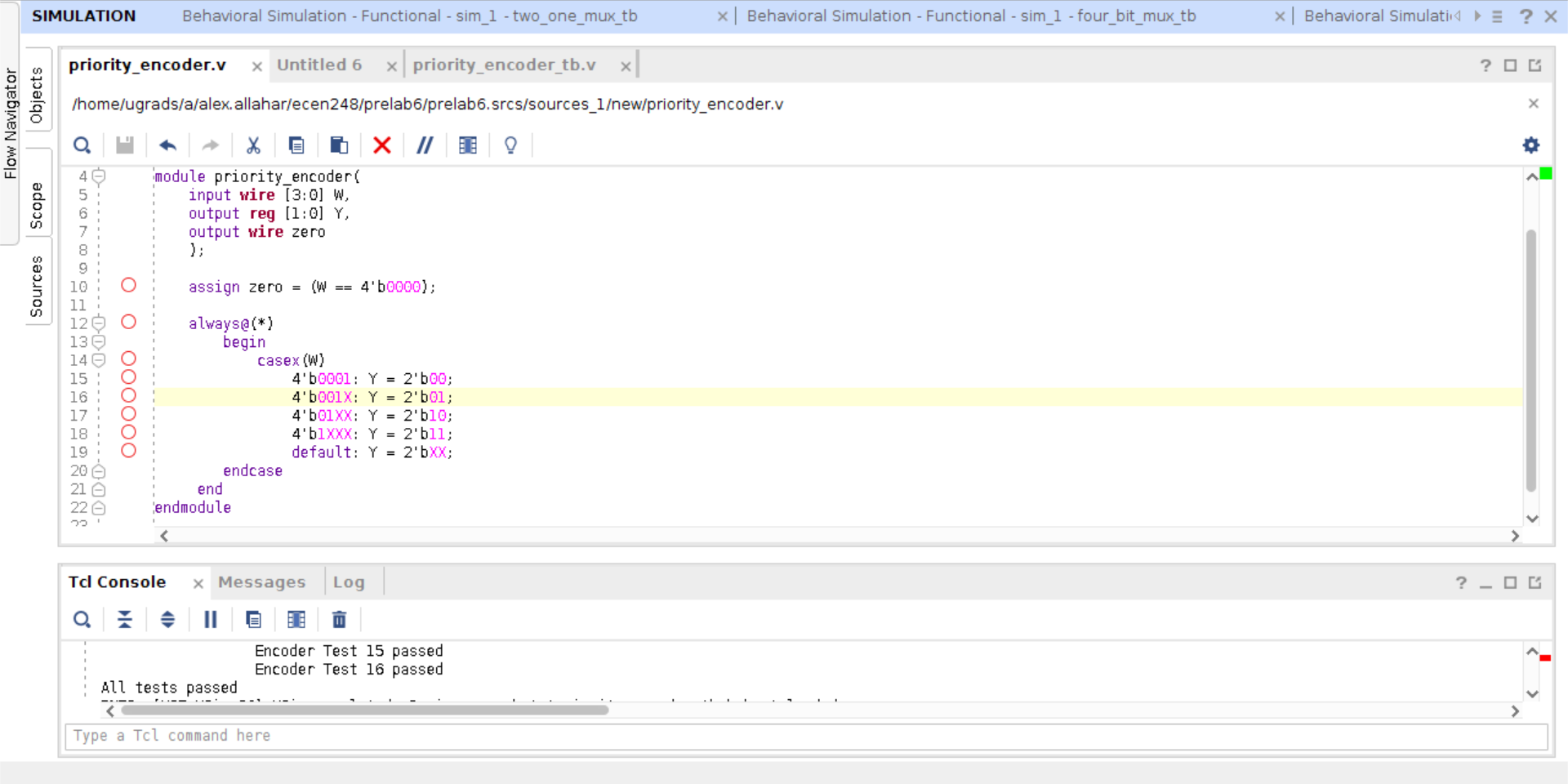
**Figure 2: 4 bit 4 to 1 code**

****

**Figure 3: 2:4 decoder code**

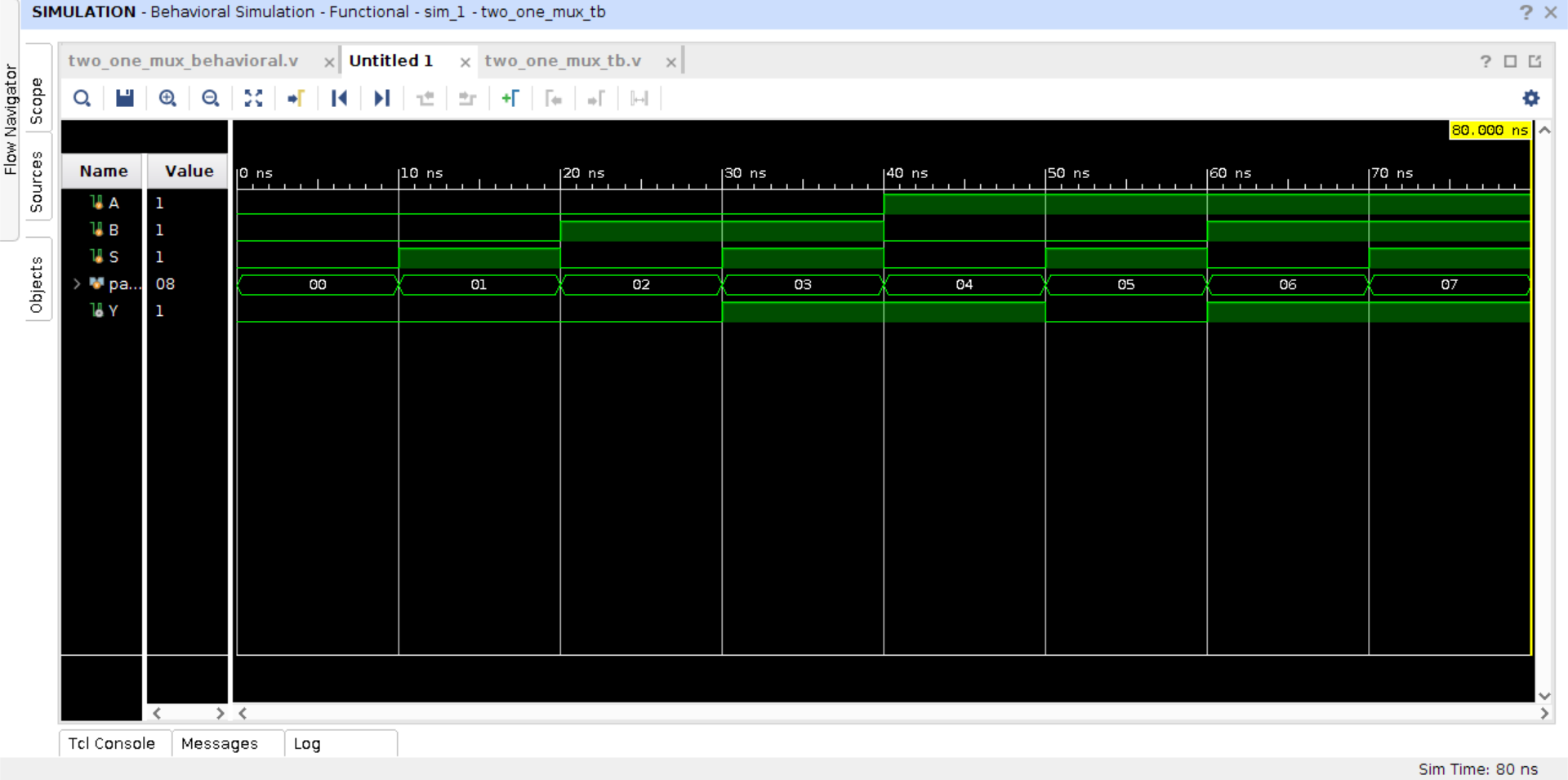
****

**Figure 4: 4:2 encoder code**

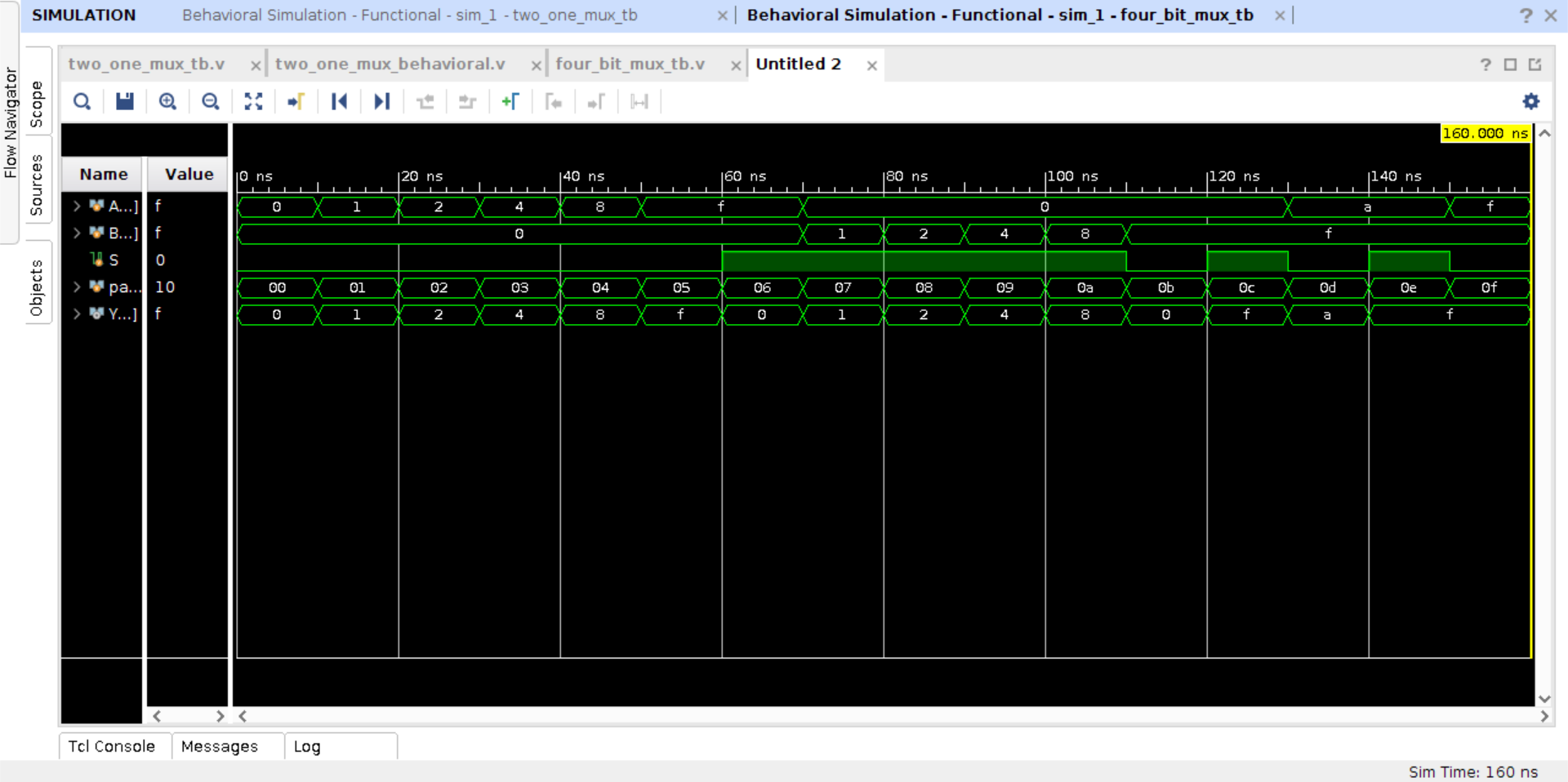
****

**Figure 5: priority encoder code**

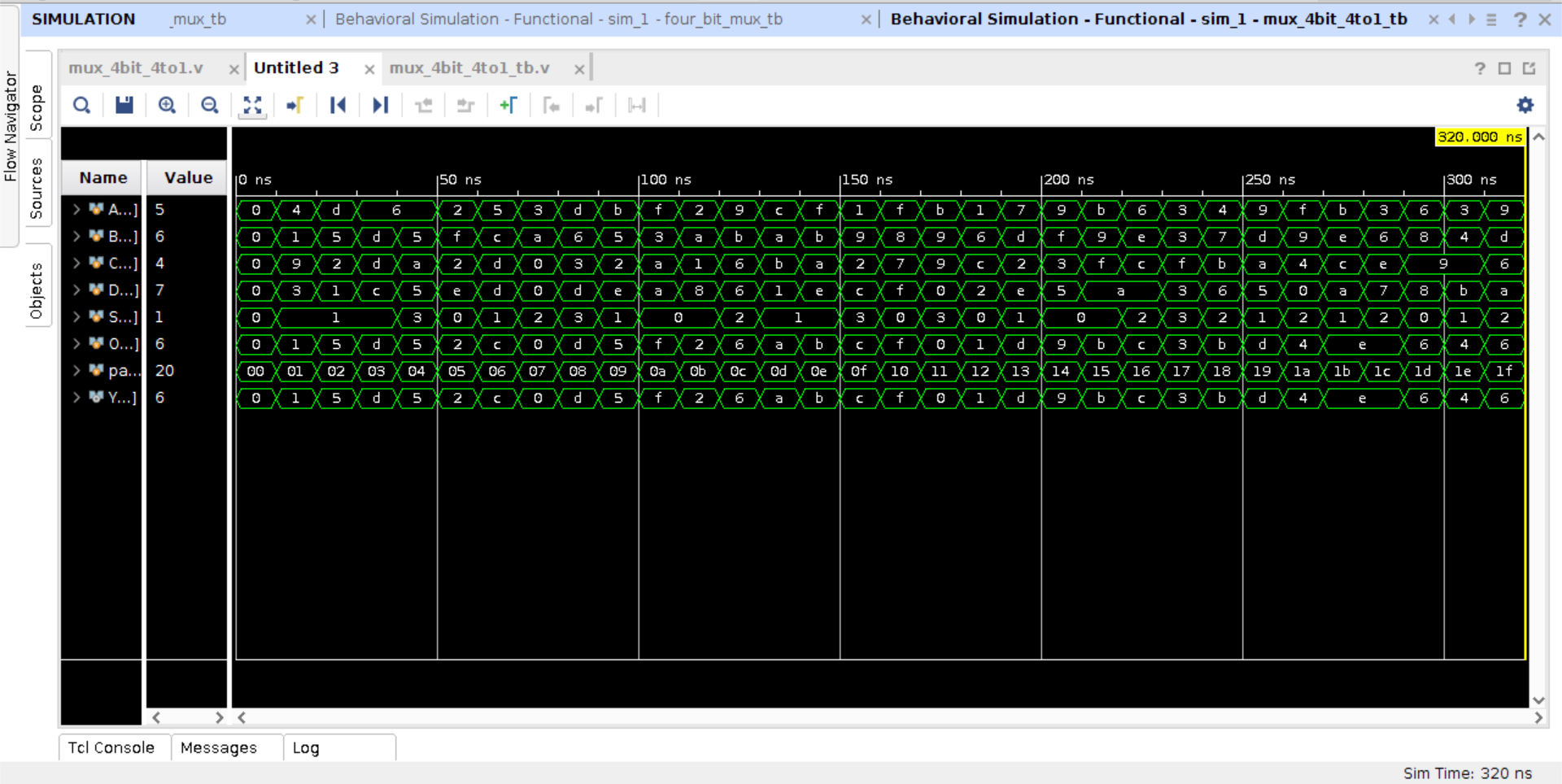
1. **Waveforms**

****

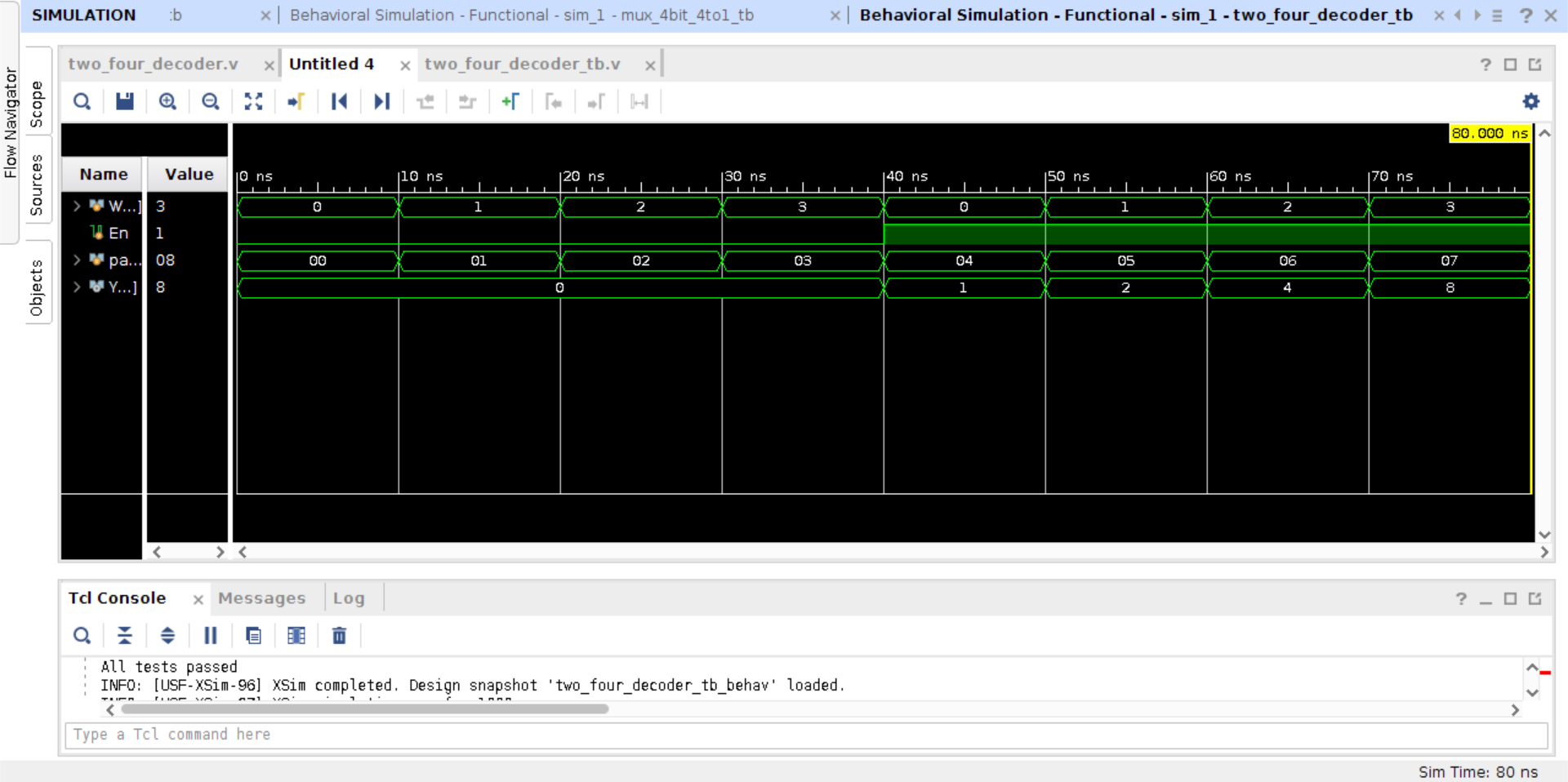
**Figure 6: 2:1 mux waveform**

****

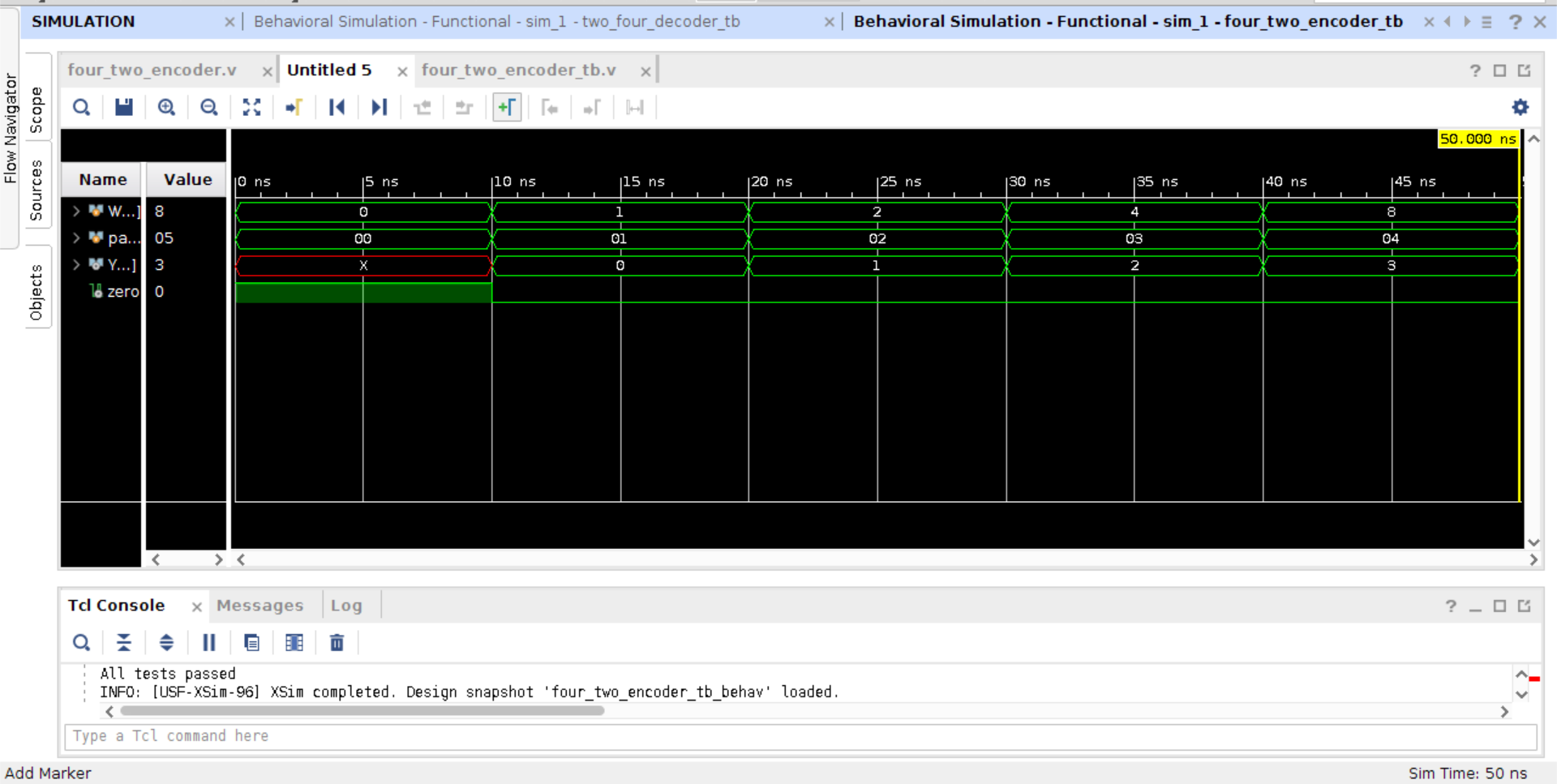
**Figure 7: 4-bit 2:1 mux waveform**

****

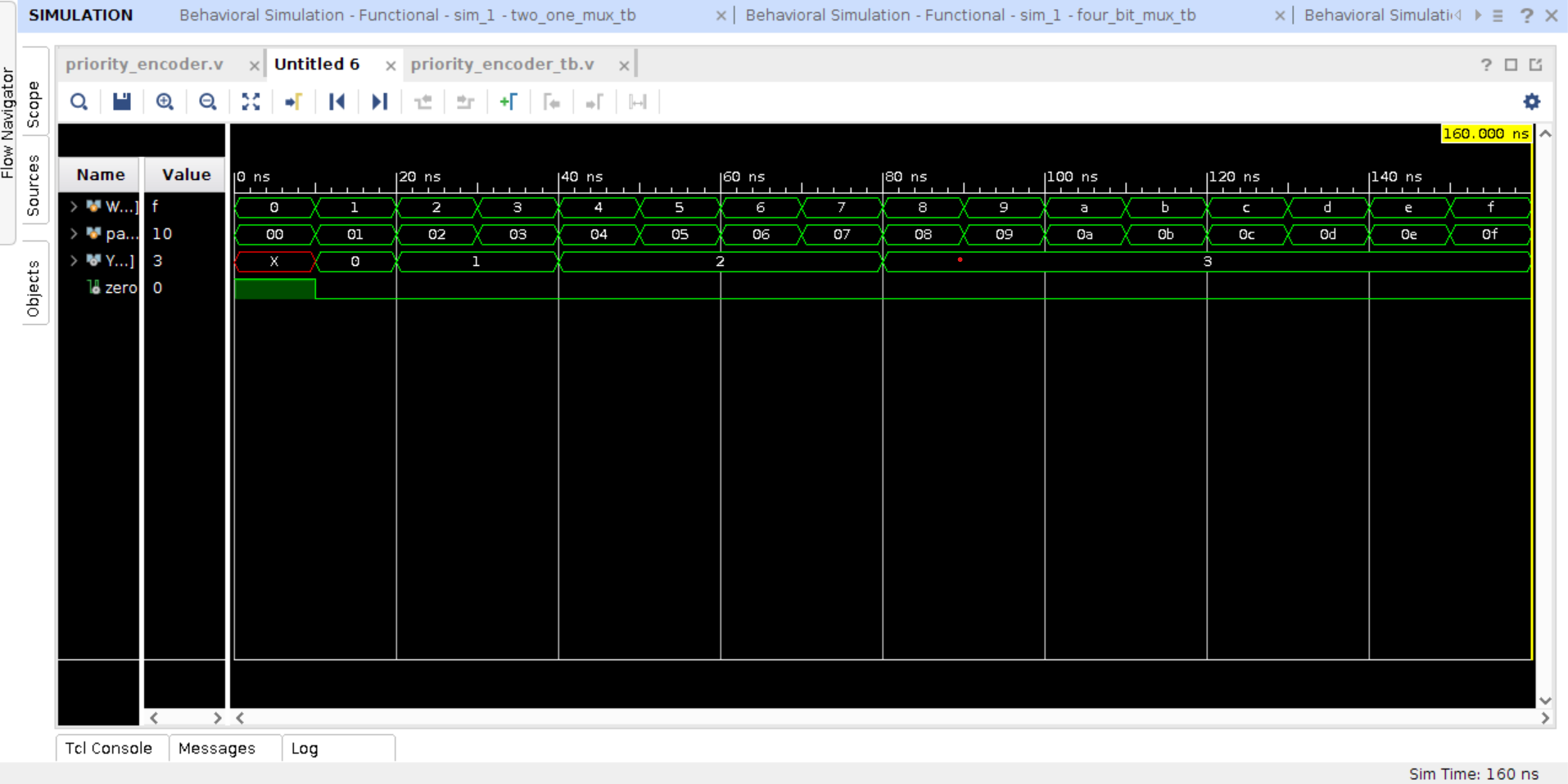
**Figure 8: mux\_4bit\_4to1 waveform**

****

**Figure 9: 2:4 decoder waveform**

****

**Figure 10: 4:2 encoder waveform**

****

**Figure 11: Priority Encoder waveform**

1. **Provide a comparison between the behavioral Verilog used in this week’s lab and the structural and dataflow Verilog used in last week’s lab. What might be the advantages and disadvantages of each?**

Structural and dataflow Verilog describes the circuit as if you were connecting various parts onto a breadboard, using basic logic gates to design the function. On the other hand behavioral modeling takes a more mathematical approach, attempting to describe how to go from input to output using loops and cases for example. Both have their pros and cons. Structural and dataflow are useful if you have a circuit schematic diagram as you can design with the digital logic gates. At the same time, behavioral modeling is more useful when the function of the course is more clear, whereas the circuit design may not be as flushed out.

1. **Compare the process of using a breadboard to implementing a digital circuit on an FPGA. State some advantages and disadvantages of each. Which process do you prefer?**

Breadboard has a lot of advantages, as one can visualize and understand the circuit design from a very structural level. However, due to the many parts and physically of the design it has a lot of potential for human error, and also cannot be shared as easily as a Verilog file. Once a Verilog file can generate a bitstream, that project can be shared and regenerate a new bitstream and uploaded to another FPGA. Furthermore, a breadboard has physical limitations whereas the Verilog design does not have the same physical restrictions. Allow for bigger designs. Thus the digital circuit implementation on an FPGA is a better process.