

Part 1)

In this exercise, you'll define a 16x8 memory block and a memory controller that will allow the user to store 8-bit values into specified memory locations and will display the contents of each register, one per second.

System Operation:

1. The system shall have a 16x8 memory block. You should implement the memory block using the skills you obtained from Intel Lab #8.
2. The system shall use the hex digits to display the following information:
 - a. HEX5 – The current write address in hex
 - b. HEX4 – Blank
 - c. HEX3 – The current read address in hex
 - d. HEX2 – Blank
 - e. HEX1:0 – The 8-bit data contained in the register pointed to by the read address, in hex
3. The system shall use the buttons KEY1 and KEY0 for the following functions when pressed:
 - a. KEY1 – The write address is incremented by 1
 - b. KEY0 – The 8-bit value selected by SW7:0 is stored into the register pointed to by the write address
4. The system shall use the switches SW7:0 to select an 8-bit value to store into the register pointed to by the write address.
5. The read address shall be incremented by 1 each second, no user action is required.

Part 2)

Write a testbench for the counter that you designed to hold the write address in Part 1.

Modify the testbench.v, testbench.tcl, and wave.do files from the ModelSim Tutorial to:

- a) Generate signals necessary to test your counter
- b) Display all input and output signals for your counter
- c) Run the simulation long enough to see the counter roll over

You should not need to do any manual adjustments to the ModelSim graphical display, the configuration files need to have the necessary information to display the waveforms correctly.