University of California, Davis

Department of Electrical and Computer Engineering

Lab 7: Simple Memory and Controller

This lab specifies the design and implementation of circuits that operate on the DE10-Lite FPGA board. All circuits are synchronously clocked by only the 50 MHz standard clock. Details not specified in this lab should be chosen by you and stated in your lab report.

I. Prelab

Complete the following and submit your work at the beginning of your lab session.

1. [5 pts] Perform and document a preliminary design including a pipelined block diagram and preliminary verilog `wire` and `reg` declarations and basic code blocks.
2. Re-read Handout *21: Overview:Memories*, taking special note of the new slides 257–259.

II. System Operation

This project uses a 4-word × 4-bit memory using positive-edge-triggered flip-flops with a synchronous read port.

System operation comprises the following phases:

1) all circuits are reset by pressing KEY0
2) after reset, the system loads the memory addresses 0–3 with data matching the address (0–3)
3) the system reads the contents of the memory in reverse order (address 3, 2, 1, 0) and displays the contents plus 1 (4, 3, 2, 1) on display HEX0 and lights LEDR9 during the four cycles the data on HEX0 is valid
4) the system loads memory addresses 0 and 1 with data 4’d05 and 4’d06 respectively
5) step (3) is repeated
6) the system goes into a wait state and lights LEDR8

Each update of the system is done exactly once per second. LEDR0 must light up the first tenth of each second after *reset* is pressed including when the system is in the final wait state.

The following values are displayed on the FPGA board:

wr_addr	HEX5
wr_data	HEX4
rd_addr	HEX3
rd_data	HEX2

III. Required Documentation

The following documentation is required and must cover the entire hardware design and all phases of operation:

1) [7 pts] Detailed pipelined block diagram
2) [8 pts] Timing diagram of all key signals
3) [5 pts] Controller state diagram

IV. Hardware Design

The design must include at least the following registers implemented as FFs:

- `state`
- `wr_addr`
- `rd_addr`

Suggestion: To update the system once each second, implement a circuit based on counter1.v from Lab 5 which pulses its output for one cycle every second.

Register (with FFs) all inputs as soon as they enter and all outputs immediately before they leave the top level module. These registers are not enable-able.

V. Testing in Simulation

[20 points] Design and write a testbench for your top-level module that exercises all functions.

For simulation, change the period of the counter so that it pulses much more often (e.g., once every 5 cycles) so a short simulation can show all phases of operation.

Once your test bench is working correctly, demonstrate it to your TA and have it checked off.

VI. Implementation and Verification on the DE10-Lite

Download your design onto the DE10-Lite board and verify it works correctly.

[55 points] Demonstrate it compiling, downloading, and operating to your TA. Your TA will record a `certutil` hash of your top-level files and this must match the hash of the files you upload to canvas so no file modifications are possible after your demo.

VII. Extra Credit

[10 points] Add an interesting capability to your design that requires a change to your datapath and pipeline (for example, enabling the memory to copy data from one location to another).

Submitted Work [100 pts total]

With the exception of instructor-provided code, all work must be yours alone.

[5 pts] **Prelab**

[75 pts] **Lab Checkoffs: Simulation and on FPGA board**

[20 pts] **Lab Report**

Submit all Verilog hardware and testbench code that you wrote. Do not include any code that you did not write such as files generated by Quartus or IP components.

a) [20 points] Required Documentation

b) Upload a copy to Canvas by performing the following steps by the end of your lab session—this is essential to receive credit for the entire lab.

1. Make a folder on your computer
2. Copy all verilog files you wrote into the folder—only the ones you wrote
3. "zip" the folder into a single .zip file
4. Log onto Canvas, click Assignments, find the correct lab number
5. Upload the .zip file

2018/05/25	Added suggestion to change counter period for simulation
2018/05/21	Added requirement to register inputs and outputs, and extra credit option
2018/05/19	Posted