Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 Introductory Digital Systems Laboratory (Spring 2009)

Laboratory 3 – Simple Memory Tester Checkoff Sheet

Student Name:_____

TA Signature/Date: _____

Must show to TA at beginning of checkoff:

- Block diagram of memory tester ☐
- Finite State Machine diagram ☐
- Verilog code printout ☐
- Memory Tester simulation waveform printout ☐

Be able to demonstrate your working lab:

- From reset, demonstrate writing and reading from different locations ☐
- Demonstrate operation with the MSB of the DATA pin disconnected (DQ[3]) ☐

Be able to respond to any of the following questions and possibly others: ☐

- What are possible problems with the address or data glitching when the write on the memory is enabled?
- Describe your timing for the memory interface.
- Can you think of a faulty circuit connection between the memory and the FPGA that will pass the test?
- What are the limitations of the simple memory test? Propose (but do not implement) a more comprehensive test.

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 Introductory Digital Systems Laboratory - Spring 2009

Laboratory 3 – Simple Memory Tester

Handed Out: 3/4/2009
Checkoff (see page 4 for details on what to turn in): 3/12/2009

Introduction

In this lab, you will design finite state machine(s) to test the functionality of a MCM6264C Static RAM (SRAM).

Procedure

This lab consists of 3 parts.

1) Wire the memory chip onto your breadboard, and connect your breadboard to the +5V, Ground, and USER4[I/O] pins of the labkit by following Figure 1. **To avoid damaging the labkit and pins, place your breadboard beside the labkit, and not on top of it. Also, please only use the labkit's +5V power supply pin, and not an external power supply.**
2) Design (block diagram and FSM diagram) and verify (using ModelSim) the tester
3) Implement the memory tester using the lab kit FPGA (code template: `lab3_labkit.v` is available from website)

For check off, be ready to demonstrate the lab and present solutions for the questions asked.

Memory Tester Wiring

The chip should be wired onto your breadboard according to the colors shown in Figure 1 (and listed on the next page), and then connected to the corresponding pins of the labkit.

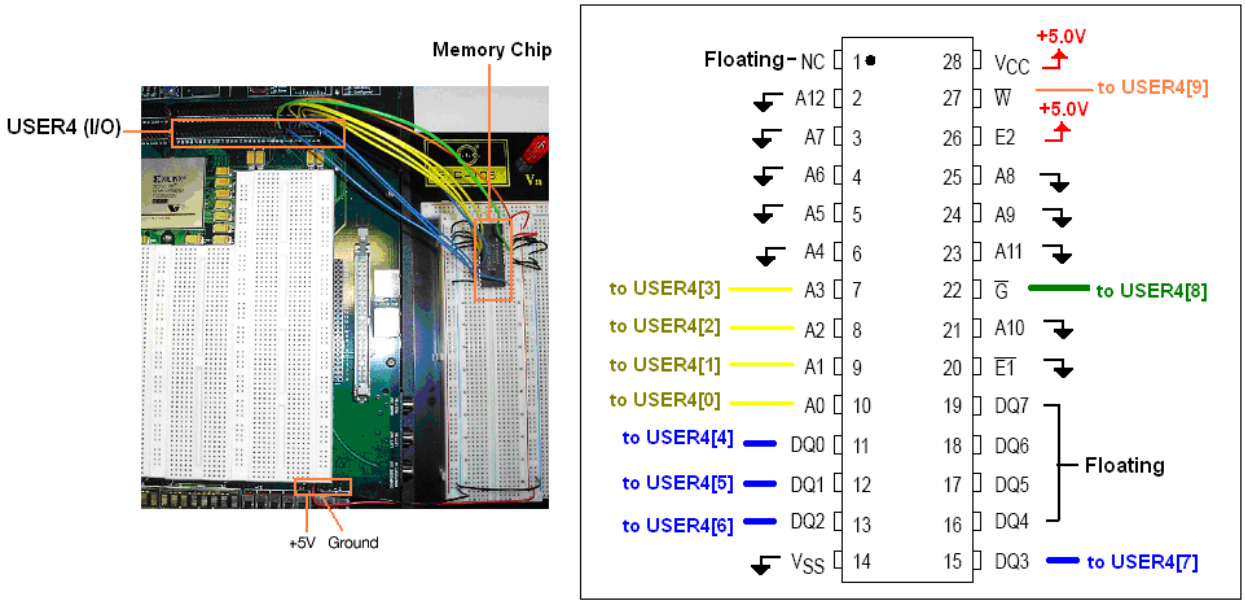

Figure 1: 6264 Placement and Wiring Diagram

Wiring Tips

1) For consistency and ease of checking, use the colors shown in Figure 1 (also given below)
   a) BLACK for signals to ground
   b) RED for signals to V_{CC} (+5V)
   c) YELLOW for all address bits to user I/O (A0, A1, A2, A3)
   d) BLUE for all data bits to user I/O (DQ0, DQ1, DQ2, DQ3)
   e) GREEN for Gbar
   f) ORANGE for Wbar
2) Remember to connect the columns of your external breadboard to the 5.0V and GND of the lab kit, as shown in Figure 1.
3) To connect to the user I/O, wire the 6264 chip to the user4 row located directly above the right side of the breadboard as shown in Figure 2.

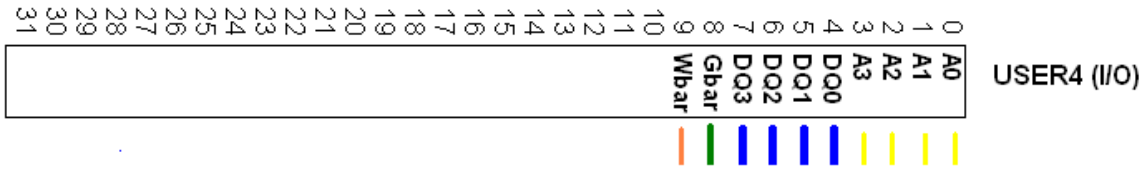

Figure 2: User I/O wiring diagram

Memory Tester Description

Ideally, a memory tester should check each bit location of the 6264 SRAM to ensure that it is accessible and interfaces correctly. In this lab, we will only be testing 16 locations. We will use the bottom 4 bits of the address space (A[3:0]) and ground all other address bits. We will use the bottom 4 data bits (DQ[3:0]) and leave the other data bits floating.

The memory tester should write the corresponding address to each location (i.e., write '0' to location 0, write '1' to location 1…write '15' to location 15), and read back the data verifying that all values were correctly written (i.e., check for '0' in location 0, '1' in location 1…'15' in location 15). It should advance to a new location once per second so that the address, data, and status can be observed on the hexadecimal display and LEDs.

As shown in Figure 3, the following should be displayed to a user:

1) `reading` LED[3] – indicates that the FPGA is reading from memory
2) `writing` LED[2] – indicates that the FPGA is writing to memory
3) `failure` LED[1] – indicates that the memory test failed
4) `success` LED[0] – indicates that the memory test passed
5) the `data` that is currently being read or written (on hexadecimal display)
6) the corresponding `address` (also on hexadecimal display)

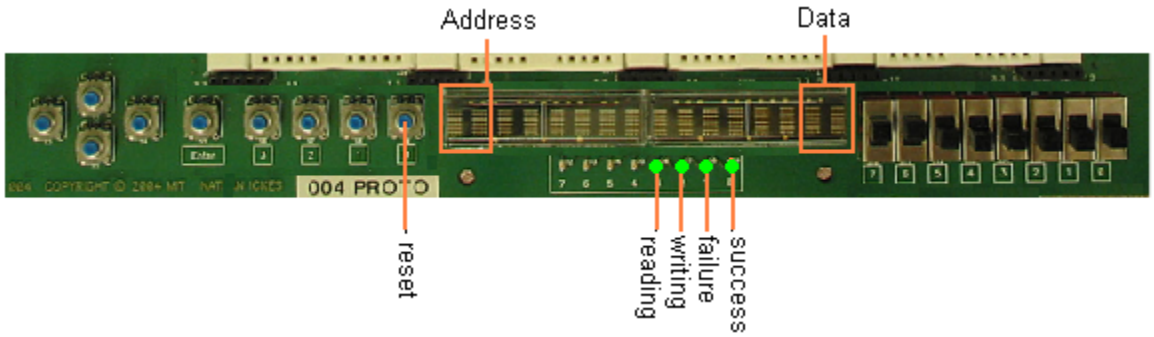

Figure 3: Memory tester user interface

The tester should be initiated from an external `reset` push button, and sequence through the locations to eventually assert the `success` LED or `failure` LED. The objective is to run the test successfully. If the test fails, the read address should stop at the failed address and the `failure` LED should turn on.

A possible block diagram is shown in Figure 4. You may use this as a general guideline for your design.

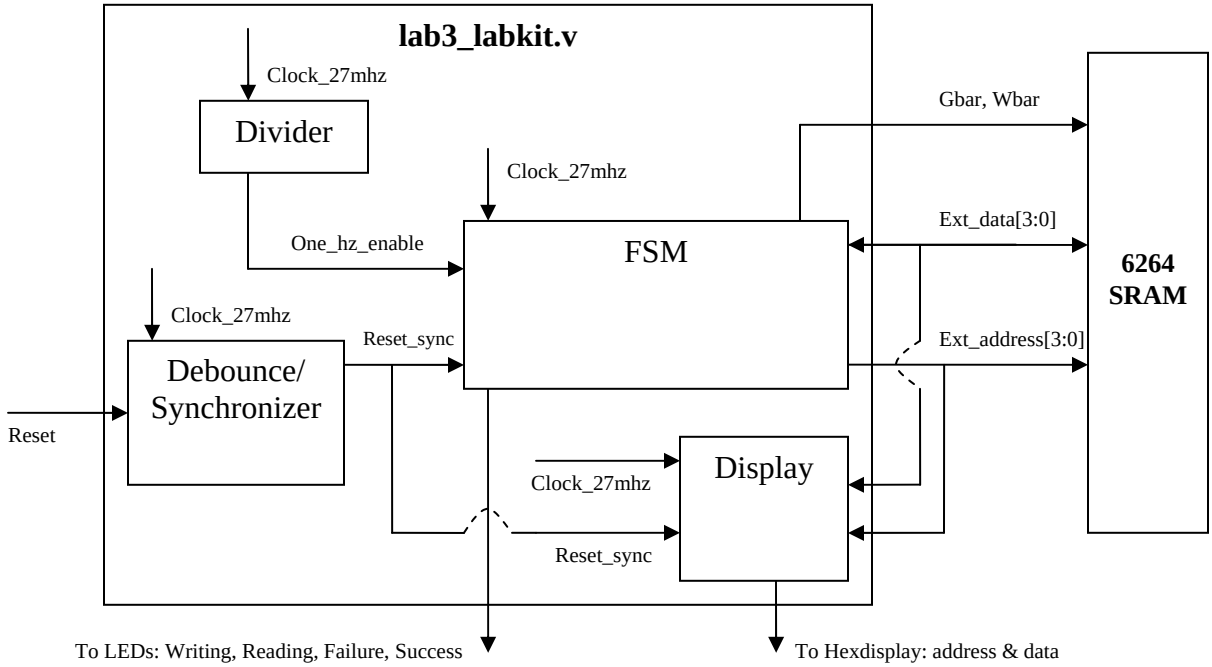

Figure 4: Overall Block Diagram

What to Turn In:

You are not required to write a detailed report for this lab. However, you must turn in the following:

1) Checkoff Sheet signed by a member of the 6.111 teaching staff
2) Detailed block diagram
3) State Transition Diagram(s) of Finite State Machine(s)
4) Verilog Code
5) Memory Tester simulation waveform printout (Use ctrl-shift-Print Screen to grab a waveform onto the clipboard, then paste it into a Word document for printout)