ECSE-323
Digital System Design

Lab #5 – ***System Integration for the Mastermind Game System***
Fall 2015

Introduction

In this lab you will put together all of the parts for the Mastermind game, and create a user-interface for the Altera board.

Learning Outcomes .

After completing this lab you should know how to:

• Get a complete digital system working on the Altera board, and will have gained experience in implementing user interfaces.

Table of Contents

This lab consists of the following stages:

1. Overview of system specifications
2. Description of the system integration procedure
3. Design of the random pattern generation circuit
4. Design of the user interface for entry and display of user guesses (to system generated patterns)
5. System scoring of user guesses
6. Generation of guesses (implementation of solution algorithm)
7. Design of the user interface for entry and display of the user generated score (to system guesses)
8. Computation of the average number of guesses needed for solution
9. Writeup of the lab report

1. System Specifications

In this lab, you will put all of the parts you have been building throughout the term into a complete working system on the Altera board. This system should have the following functions, divided into two modes (*user guessing* and *system guessing*):

User Guessing Mode

• Generate a random hidden pattern of peg colours (4 pegs with 6 possible colours). A new pattern should be generated on a user button press.
• Input a user guess as to the hidden pattern, and compute the score for that guess. The user input and the score are displayed on the four 7-segment LEDs.

System Guessing Mode

• Implement the solution algorithm. At each stage in the algorithm the system should display its current guess, and wait for the user to enter the score for that guess. In addition to the current system guess, the user-entered score should also be able to be displayed on the four 7-segment LEDs.
• The algorithm should be re-started on a user button press.

The two modes should be selected via a switch on the Altera board.

2. System Integration Procedure .

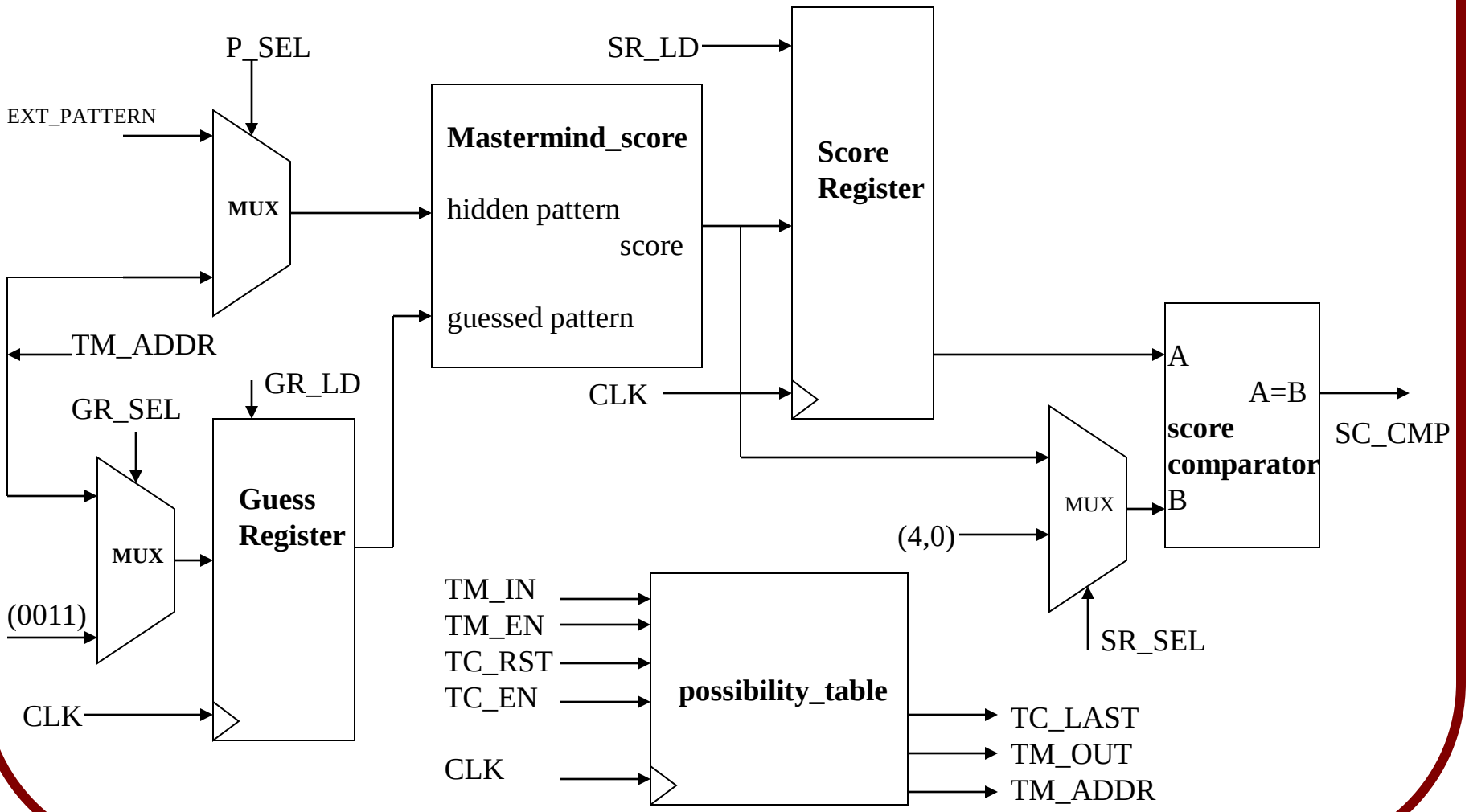
The system should be designed using a datapath/controller approach. You can use the datapath/controller designed in lab #4 as a starting point. The datapath constructed in lab #4 is shown on the next page for reference.

Rename your lab #4 controller circuit as ***gNN_mastermind_controller*** (if it does not already have that name). Put your datapath elements and the controller into the top level project schematic drawing.

In the following sections of the lab, you will add functionality to the controller and datapath, bit by bit, until the system can carry out all of the functions given in the specifications.

To begin, the first modification to the controller should be to add the mode selection. One of the switches on the Altera board should be used to select the mode. The mode switch output should be used a control input, steering the controller FSM to one sequence or another in the FSM state diagram.

Starting point for the Mastermind Datapath (from lab #4).



3. Design of the Random Pattern Generator .

In order to provide a challenging and interesting game, the system should be able to generate new peg colour patterns randomly.

The system specifications ask for 4 pegs, with 6 colours each, for a total of 6x6x6x6=1296 different possible patterns.

You could do this with a random number generator that has 12 bits but you would have to somehow avoid the invalid patterns.

Instead, you could use the inherent uncertainty induced by the user's input timing to sample the count value of a high-speed counter. That is, create a 1296-step counter that is running at 50MHz. Use the output of a push button to enable a latch connected to the output of the counter. When the push button is released, the latch will hold the last count value. This will be effectively random, since the user cannot control his/her button pressing at a nanosecond level!

As in lab #3 you want the counter to have 12 bits of output, where the bits are arranged into 4 groups of 3-bits each, with each of the 3-bit groups cycling through the counts 0,1,2,3,4,5,0…

So, add the table counter that you constructed in lab 3 to the Mastermind game datapath that you constructed in lab #4, and connect its output to a 12-bit register. The load enable for this register will now be another control signal to be provided by the system controller. This counter can be left enabled all of the time.

Add to the controller a sequence of states that is reached when the mode input is high (this will serve to select the "*User Guessing*" mode). The other sequence of states (corresponding to the lab #4 controller activity) will be reached when the mode input is low (i.e. in "*System Guessing*" mode).

For now, the *User Guessing* mode controller behaviour will be quite simple - it enables the register load enable when one of the push-buttons on the Altera board transitions from high-to-low. ***You don't have to worry about switch bounce in this situation (why not?)***. So your controller should wait for the push-button to go high, then wait for it to go low, at which point the latch load should be enabled for one clock cycle, then return to waiting for the push-button to go high again.

The register will have four 3-bit groups, each corresponding to the colour of a peg in the generated pattern. For testing purposes it will be useful to see the value of these colours on the 7-segment LEDs.

Demonstrate the random pattern generator to the TA.

TIME CHECK

You should be this far at the end of your *first* 2-hour lab period!

4. Design of the User Interface for Guess Entry and Display.

In order to make this game usable, there must be some way for the user to interact with it. You will construct the user interface step-by-step.

The first step will deal with the mechanism by which a user can enter his own guesses as to the identity of a system-generated hidden pattern.

Modify the current system datapath (by adding multiplexers and registers as required) and controller, that will add the user guess input and display capability.

Demonstrate the user guess input display capability to your TA.

5. System Scoring and Display of User Guesses.

The next step in the user interface involves the design of the part of the system that compares the user guess to the system-generated hidden pattern and generates, and displays, the score.

Modify the current system datapath (by adding multiplexers and registers as required) and controller, that will add the user guess scoring and display capability.

Demonstrate the user guess scoring display capability to your TA.

At this point, a user should be able to play the Mastermind game in *User-Guess* mode against the system generated hidden patterns. Try a few rounds.

The *System-Guess* capability will be added in the next few sections of the lab.

TIME CHECK

You should be this far at the end of your *second* 2-hour lab period!

6. Integration of Solution Algorithm.

You will now modify your system to permit operation in "System-Guess" mode.

You already have most of the controller FSM implemented for this mode, using the controller from lab #4. But you need to add the capability for the system guesses to be displayed on the Altera board LEDs. You also have to add the "Restart" function, which will initialize the solution algorithm.

Modify the datapath (by adding multiplexers and/or registers where appropriate) and the controller to allow the system generated guess to be displayed on the 7-segment LEDs (with one LED per peg).

Demonstrate the system guessing and display capability to the TA.

TIME CHECK

You should be this far at the end of your *third* 2-hour lab period!

7. Design of the user interface for entry and display of the user generated score.

In order for the system guessing algorithm to function, it requires feedback from the user in the form of a user-generated score.

So, modify the datapath and controller to allow the user to enter the score, and to display the user input.

Demonstrate the user input and display capability to the TA.

At this point, your system should be completely functional!
Play with it and make sure that it is operating correctly.

8. Computation of the average number of guesses needed for solution.

Once you have your system completely functional, try to obtain an estimate for the average number of guesses needed by the system to find the solution. Do this by running the system for at least 10 trials and averaging the number of guesses for each trial.

If your system is working properly, the average should be less than 5!

Show your results to the TA, by running through a few trials.

TIME CHECK

You should be this far at the end of your *fourth* 2-hour lab period!

9. Writeup of the Lab Report .

Write up a report for the complete ***Mastermind Game*** system that you designed. You do not need to give details of modules that you designed in labs 1 through 4. Just make reference to their reports as needed.

The report must include the following items:

- A header listing the group number and the names and student numbers of each group member.
- A title, giving the name of your system.
- A description of the system's features (like an advertising brochure describing what the system does).
- A block diagram of the entire system.
- A *detailed* description of how your system works, referring to the block diagram.
- A description of the user interface (i.e. a users guide to operation).
- A discussion of how the complete system was tested.
- A summary of the FPGA resource utilization.
- A conclusion section discussing problems or significant issues that arose during the design process, as well as a discussion of possible enhancements or extensions that could be made to your system.

Some items to remember when preparing the report for lab #5:

- The document must have page numbers.
- All diagrams, tables, and figures should have captions describing the contents of the figure, and be given a figure number.
- All such figures must be referred to in the text of the report. Do not just throw a figure into the report without referring to it. Figures should be used to augment the textual matter.
- If you include figures or text taken from other sources, be sure to cite these properly. **Avoid plagiarism!** All VHDL descriptions should contain the name of the designers and the date written.
- Hand-drawn figures and diagrams are not acceptable!
- Turn off the grid in the Quartus schematics when making screenshots!
- Break large block diagrams into multiple smaller ones when inserting them into the report.
- Do not include long vhdl descriptions into the report. Instead, include the .vhd files in your report submission zip file.

Don't wait until the last day to start writing the report. Start writing it early on in the lab 5 period. *The report is due on the last day of classes*, and not one week later!

The report should be done in html or pdf (preferred), or in Microsoft Word, and uploaded to the *myCourses* site using the lab #5 assignment submission icon.

The presentation of the report (grammar, spelling, and clarity of the diagrams) will also be graded.

Make sure that you have uploaded ***all*** of the design files (e.g. .bdf and .vhd files) used in your project. Also include the device programming file (.sof or .pof file) so that the grader can download your design to his/her Altera board.

The report is due at midnight on the last day of classes, *Monday, December 7*. Late submissions will be accepted, but will be assessed a penalty of *1 mark per day*, (out of a possible maximum of 10).

Grade Sheet for Lab #5

Fall 2015.

Group Number: _____.

Group Member Name: _____. Student Number: _____.

Group Member Name: _____. Student Number: _____.

Marks			TA Signatures
	1.	Demonstration of the random pattern generation circuit	.
	2.	Entry and display of user guesses for system generated patterns	.
	3.	System scoring of user guesses	.
	4.	Generation of guesses (implementation of solution algorithm)	.
	5.	Entry and display of user score (to system guesses)	.
	6.	Computation of average number of guesses needed for solution	.

Each part should be demonstrated to one of the TAs who will then give a grade and sign the grade sheet. Grades for each part will be either 0, 1, or 2. A mark of 2 will be given if everything is done correctly. A grade of 1 will be given if there are significant problems, but an attempt was made. A grade of 0 will be given for parts that were not done at all, or for which there is no TA signature.

All demos must be done by the last session on Monday, Dec. 7. There will be no extensions given for demos.