Lab 2: The Price is Random

Due Friday, October 26 2018, 11:59 PM

Minimum Submission Requirements

- Create a Lab2 folder (note the capitalization convention, include no extra characters in the directory name) that contains the following files:
    - Lab2.lgi in the Lab2 folder
    - README.txt
- Commit and Push your repository
- Tag the commit that you would like to be graded
    - The tag must be in the form Lab2_submission_# (note the capitalization convention)

Lab Objective

In this lab you will design and simulate a guessing game that compares a user input to a random number. If the user input matches the random number, the score is incremented by 2; if the user guesses low, by 1, and if the user guesses high, by -1.

Description

A random number will be generated and compared against a keypad entry. Seven-segment displays will be used to display the user answer (keypad entry), the correct answer (random number), and the score.

Use two's complement to keep track of a negative score. Numbers shown must be human readable, i.e. if the value is negative, the score must be shown with a negative sign. (Hint: If the number is negative, display the additive inverse and a negative sign.)

Top Level

Here is a top-level block diagram of the game. Note that clouds indicate combinational logic.

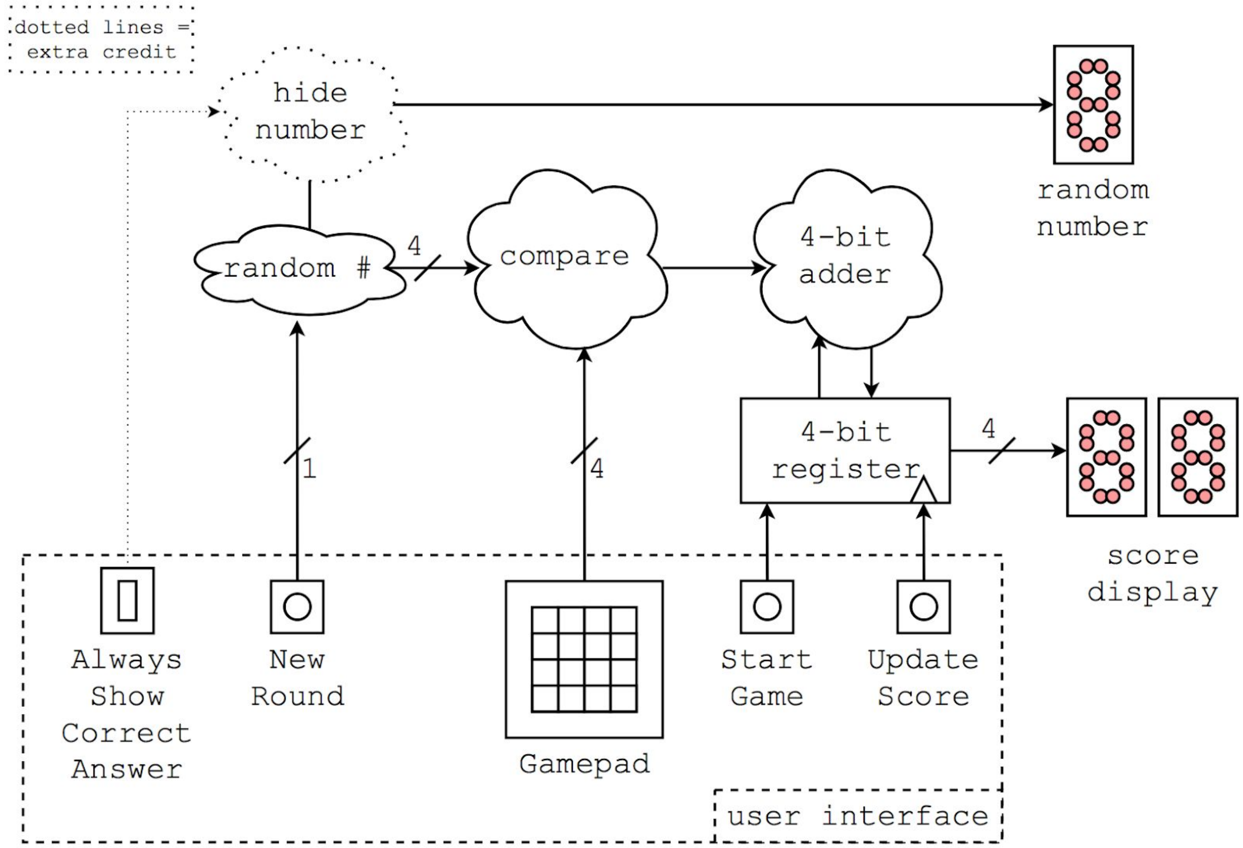


Figure 1: Block diagram depicting price is random game

Note

For the register, use D flip-flops, and make sure they are edge triggered with a clear line. You may NOT use the counter, mux or ALU objects provided in MML.

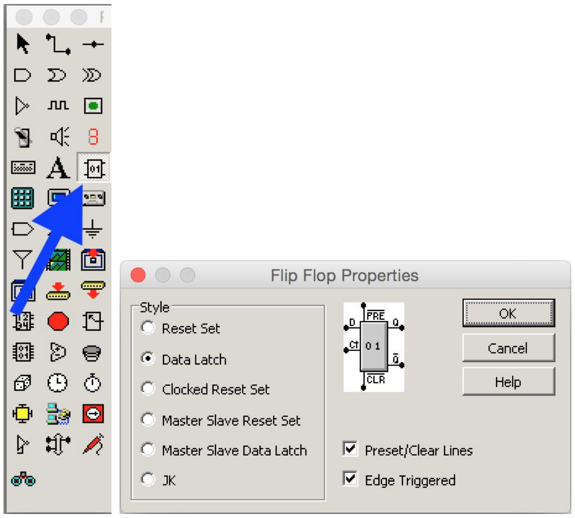
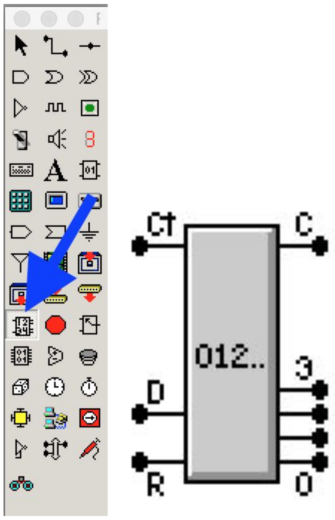
USE	DON'T USE
 Flip Flop Properties Style ○ Reset Set ● Data Latch ○ Clocked Reset Set ○ Master Slave Reset Set ○ Master Slave Data Latch ○ JK ☑ Preset/Clear Lines ☑ Edge Triggered OK Cancel Help	

Figure 2: Flip Flops in MML palate

For your convenience, here is a tabular description of how flip-flop with clear line works. You may also find practice_flipflow.lgi handy for understanding how flip-flops set and reset.

Extra Credit

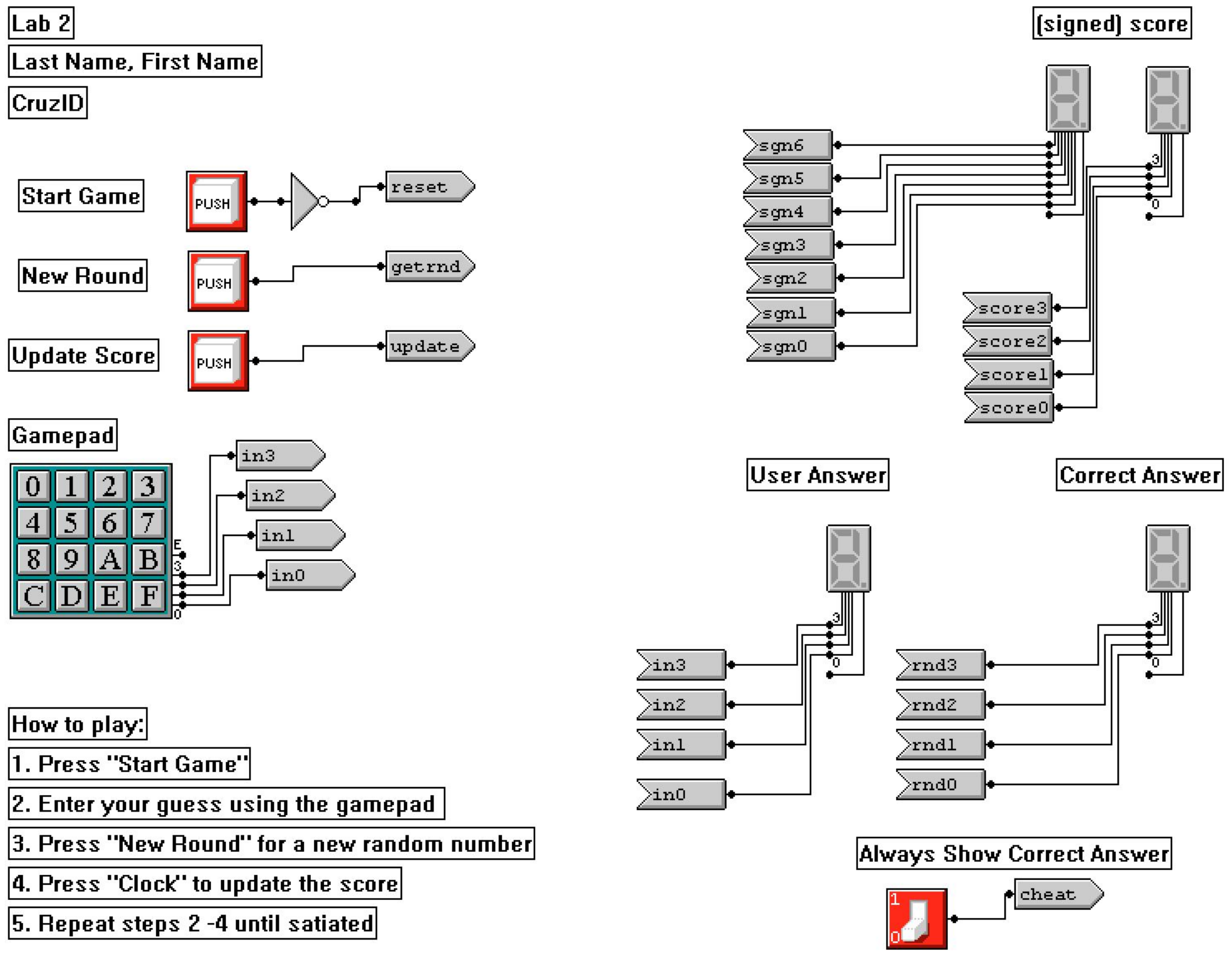
You may have noticed, with the current implementation, you can cheat by simply pressing the keypad after the random number is generated and displayed.

1. Add functionality to prevent users from cheating in this fashion.
2. Make the “Always Show Correct Answer” switch work. Make it so when the switch is off, the seven segment display for “correct answer” will display “0.” (zero with a decimal point). If the switch is on, it will show the random number (no decimal point). Hint: think about muxes.

If you choose the extra credit route, add a comment on the first page of the template indicating you have chosen the extra credit option.

Template

The components on the first sheet of your schematic file should match the template. You are permitted to change only the text - your name, CruzID, and the sender and receiver names if you’d like. Additional wires and logic circuits shall be drawn on subsequent pages. If you are not choosing the extra credit option, remove the switch labeled ‘cheat’ and the sender associated with it. You may add as many pages as necessary.



How to play:

1. Press "Start Game"
2. Enter your guess using the gamepad
3. Press "New Round" for a new random number
4. Press "Clock" to update the score
5. Repeat steps 2 -4 until satiated

Figure 3: A template showing the overall components setup of lab2

An example of how to hook up the D flip-flop is shown on page 2 of the template. This circuit should be deleted in the version of the lab that you turn in.

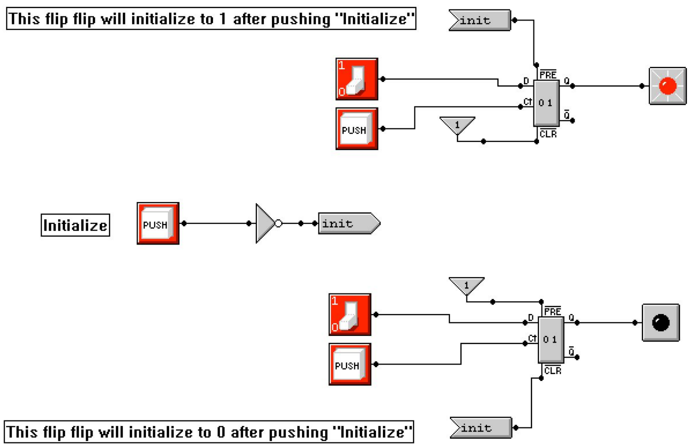

This flip flip will initialize to 1 after pushing "Initialize"
init
PUSH
Initialize
PUSH
init
PUSH
This flip flip will initialize to 0 after pushing "Initialize"
init

Figure 4: Flip-flop usage example

Comments

Each page should be labeled with your last name, first name, and CruzID (the name used in your UCSC email address). Label each circuit with a description of the functionality and the part of the lab that they are for.

Visual Structure

Presentation of information is an important part of deliverables. Clean documentation is easy to comprehend and looks professional. Your circuits should be structured in an organized method that is easy to read and interpret. Using the “Snap to Grid” setting under the View menu makes it easy to line up components. Do not have crossing wires.

README.txt

This file should be a plain text document and contain your last name, first name, and CruzID. Use the template shown below (also on canvas). Put effort into your answers.

```
------------------------
LAB 2: The Price is Random


CMPE 012 Summer 2018

Last Name, First Name
CruzID
-------------------------

What were the learning objectives of this lab?
Write the answer here.

Did you encounter any issues? Were there parts of this lab you found enjoyable?
Write the answer here.

How would you redesign this lab to make it better?
Write the answer here.
```

Missing Wire Best Practices

MML has a known bug which causes some wires to disappear after reopening the file. To reduce the likelihood of this occurring, DO NOT use the “Node” tool (it’s a tiny black dot located at the top-right of the tool palette). This tool is particularly vulnerable to the bug.

If this bug occurs, the grader will attempt to repair the missing wire in your file. This is only possible if your circuit is very readable. Make sure that wires do not cross whenever possible. Wire paths should be short and direct. Use receivers very liberally.