Lab 2: Game Access Control on FPGA

Stefan Bucur 3153

ECE5440

Contents

[**1.** Introduction 3](#_Toc32790398)

[**2.** System Architecture and Design 4](#_Toc32790399)

[2.a: Button Shaper (bshaper) 4](#_Toc32790400)

[2.b: Load Register (loadreg) 5](#_Toc32790401)

[2.c: Access Control (access) 6](#_Toc32790402)

# **1.** Introduction

Lab 2 is a math memorization game that tests players’ ability to add and convert between binary, decimal and hexadecimal in their minds. The development in this lab focuses on implementing a login system to the game and a more robust player input system. When the device is first powered on, the number displays and indication lights are visible, but the player(s) cannot immediately begin playing the game; they must first enter the correct sequence of four-bit numbers one by one into the login system. Once the correct sequence is read, the system will indicate the game can start and players can begin entering numbers with their designated button and sets of switches. Players enter numbers into the system by toggling switches that represent a binary number and pressing their button. If the two numbers add to 0xF (15 in decimal), the FPGA will change the LEDs to indicate a correct result. Players record score using their own means and can take turns being the first player to enter a number that round. If the players would like to leave the board for any reason and not want others to alter the inputs, they can log out of the game and the system will preserve the present values until the players log in again. At any point during operation the reset button can be pressed, reverting the system back to its initial state as if it was powered on for the first time. In the modules with a state machine, the transition upon activating RESET is assumed to be its default state.

# **2.** System Architecture and Design

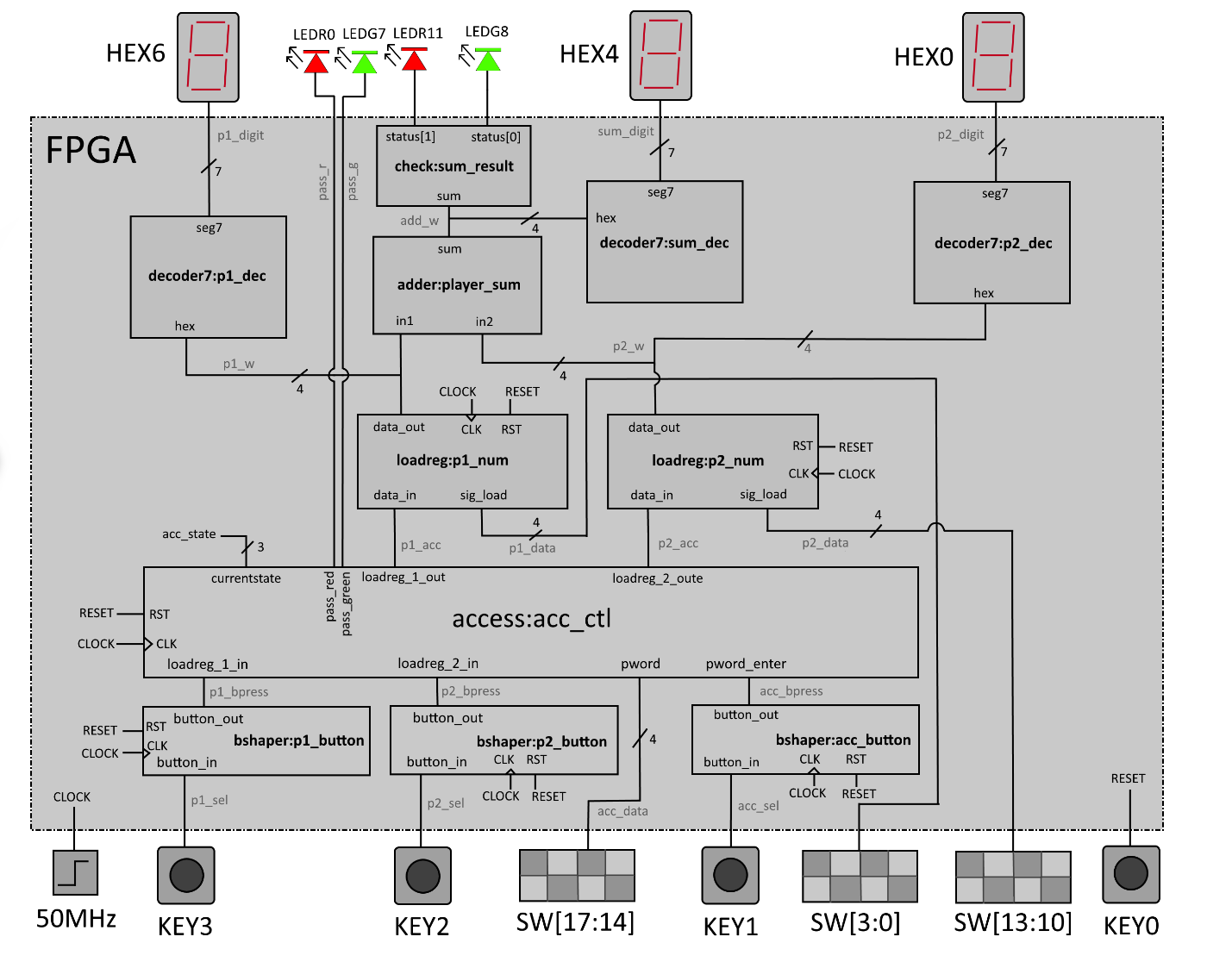


Figure : FPGA system architecture for Lab 2. KEY inputs pin out to the 4 onboard buttons. SW switches pin out to the onboard switches. A 50MHz clock is wired into the FPGA from an onboard oscillator.

Overall, the system’s sequential logic modules use the DE2-115’s 50MHz internal clock signal to update on its rising edge signal. Apart from RESET, the other three input button signals are first passed through their own button shaper module. This is due to the internal state machines of both the Access Control and Load Register modules treating a button activation signal as a new button press upon each new clock cycle. In order to reduce the complexity of the state machine, the button shaper converts the button’s debounced active low signal into a positive logic pulse that lasts one clock cycle. RESET is not shaped because the sequential logic modules should ideally be subject to multiple subsequent reset signals to ensure the entire system is set to the same initial state. The acc\_state wire within the FPGA is left for debugging purposes in simulation. Explained further below is a detailed explanation of the functionality of each of the modules in Figure 1.

## 2.a: Button Shaper (bshaper)

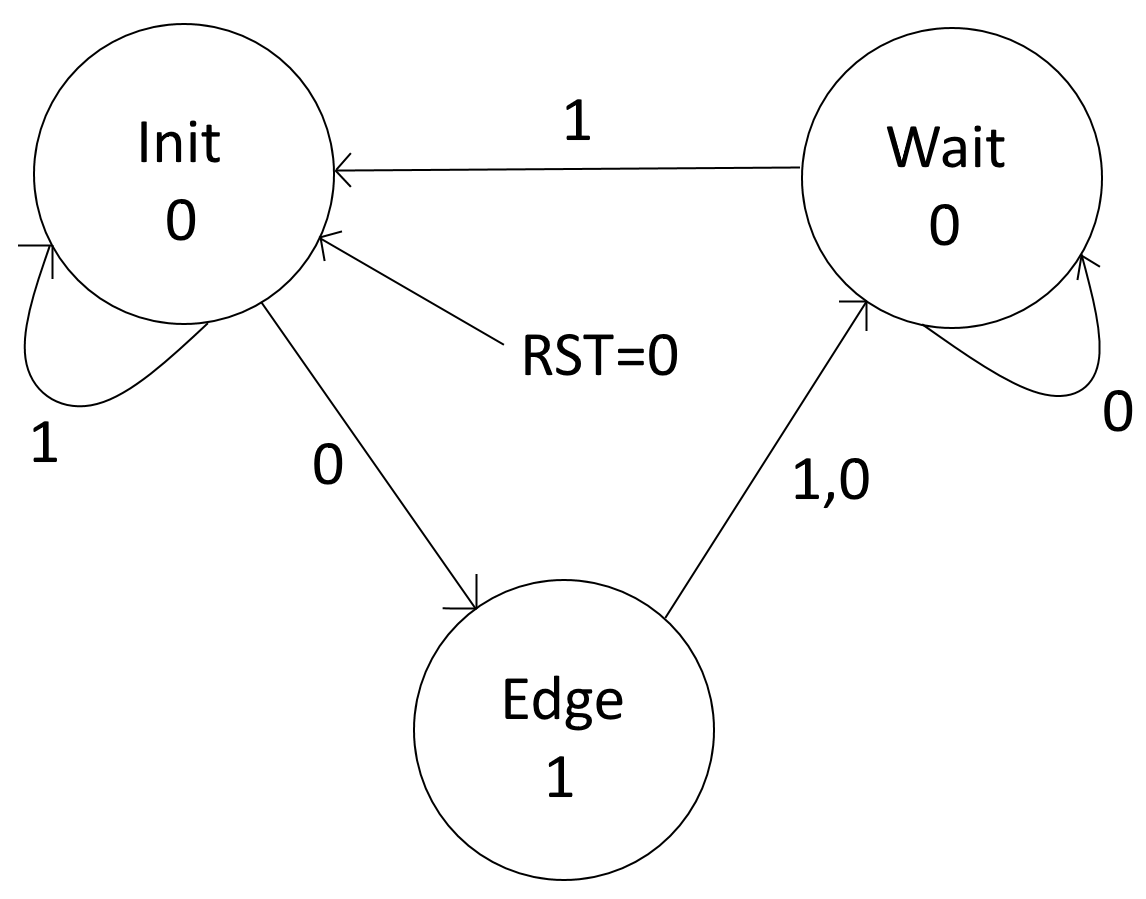


Figure : Finite state machine for the button shaper. State values correspond to the button\_out signal. Transition arrow values correspond to the button\_in signal. RST = 1 does not affect the FSM and is therefore omitted.

The button shaper module converts an active low input signal that can last multiple clock cycles and converts it into an active high square pulse lasting only one clock cycle. The functionality is implemented using a two-procedure state machine. The Init state is the default state, where it waits for the button active signal to change at which point it will transition to the edge state within that clock cycle. At the edge state, the output signal is set high and the state machine transitions to a wait state. Here it waits for the input signal to revert to its inactive value at which point the button shaper returns to the Init state. If RST is active when the button is held down, the state machine will ignore any button input until the reset is deactivated. Note in the simulation how the second output pulse occurs at the rising clock edge that proceeds the reset signal being turned off.

## 2.b: Load Register (loadreg)

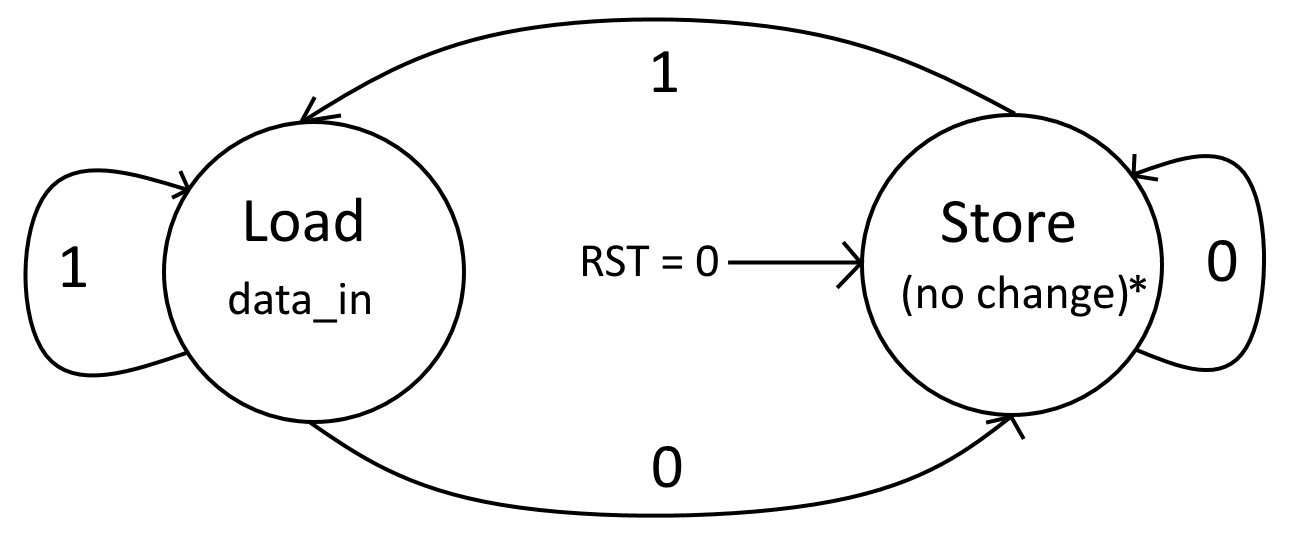


Figure 3: Finite state machine for the load register module. State values correspond to the data\_out signal. Transition arrow values correspond to the data\_in signal. RST = 1 does not affect the FSM and is therefore omitted.

## 2.c: Access Control (access)

## 2.d: Addition (adder)

## 2.e: 4-to-7 decoder (decoder7)

## 2.f: Check Module (check)

# **3.** Simulations