

CSE 490/590 Computer Architecture - Spring 2020

Project-2

Due: Friday, May 8th, 2020 at 11:59 pm

Project Choices:

There are multiple choices for your project on the following pages. You should choose 2 of them as preferred and alternate and submit your choices. You will then be assigned one of those 2. Projects are for individuals.

Instructions for submission of proposals and choices, as well as the project files will be provided.

Pay attention to the instructions and deadlines. Emails are not accepted.

Documentation and Submission Requirements:

Your submission should contain all the source files needed to synthesize and demo your project. You must also submit a design document in PDF format. Your design document should include your design idea, any extra detail that is not defined by the project description, breaking down of your design and step-by-step implementation, and mistakes that you learned from.

Electronic Lock

Design an electronic combination lock. A four digit code ($D_3D_2D_1D_0$) can be entered by the push buttons and slider switches. Slider switches 0 through 3 are used to enter a BCD, while the momentary push buttons are used to latch the value on the switches. The left push button (L) is used to latch D_0 , the upper push button (U) is used to latch D_1 . The lower push button (D) is used to latch D_2 . The right push button (R) is used to latch D_3 .

When the entire code has been entered, the center push button (C) should be pressed. When the correct combination is entered and the device unlocks, the LEDs will flash. If two attempts are made to unlock the device without correctly entering the combination, the device should be deactivated for 10 seconds.

The seven segment display will be used to display the state of the lock. The states are: "LOC" (locked), ##### (Latched digits during an attempt), "UnLC" (Unlocked), "PAUS" (deactivated), or "CHnG" (Changing combination).

For each unlock attempt, the entire code must be entered within 30 seconds (from the first button press to the last button press), or the attempt resets and seven segments show "LOC" again. Note that this does not count towards incorrect attempts.

When the device is unlocked, it can be locked again by pressing the center momentary push button (C), or the user can change its combination by pressing the right momentary push button (R). After pressing the button, the seven segments should show "ChnG", and the same procedure for unlocking is repeated. After changing the combination, the device should immediately go to locked state.

The default combination for your lock should be the last 4 digits of your UB person number. For example, if your UB person number is 50429876, then your combination would be 9876.

Guessing game

Design a two player guessing game. The game starts with the four digit seven segment display illuminated with "PL 1" indicating that player #1 should enter a number between 0x0000 and 0xFFFF. The number is entered using the push buttons and slider switches. After the number has been entered, player #1 should press the center momentary push button (C). The seven segment displays are used to indicate this by now displaying "PL 2", indicating it is player #2's turn. Player #2 repeatedly enters four digit numbers until they correctly guess the number entered by player #1. The procedure for entering the four digit number ($D_3D_2D_1D_0$) is as follows: **Slider switches 0 through 3 are used to enter a hexadecimal digit (DO NOT use all 16 switches)**, while the momentary push buttons are used to latch the value on the switches. The left push button (L) is used to latch D_0 , the upper push button (U) is used to latch D_1 . The lower push button (D) is used to latch D_2 . The right push button (R) is used to latch D_3 . When the entire number has been entered, the center push button (C) should be pressed. When an incorrect number is entered, the display should indicate if the guess was too high ("2 HI") or too low ("2 LO"). When the correct guess is entered the LEDs should blink in celebration, and the number of guesses required should be displayed on the seven segment display. The number of guesses should be displayed as a decimal number (NOT hex).

The seven segments should show the latched digits during the guessing procedure as well as DP segments to show the number of incorrect guesses (example # # #.#. Indicates the current guess value and that the guesser has 2 incorrect guesses). Initially all of the DP are off, after each incorrect guess, one of the DP lights up and remains on until the end of the game. If 4 incorrect guesses are made, the seven segments should show "LOSE" for 3 seconds, and the game should be reset.

Calculator

Design a calculator. All operands are decimal numbers. The operands are entered using the push buttons and slider switches. The four digit operand ($D_3D_2D_1D_0$) is entered as follows: Slider switches 0 through 3 are used to enter a BCD digit, while the momentary push buttons are used to latch the value on the switches. The left push button (L) is used to latch D_0 , the upper push button (U) is used to latch D_1 . The lower push button (D) is used to latch D_2 . The right push button (R) is used to latch D_3 . When the entire operand has been entered, the center push button (C) should be pressed. If the user enters a digit greater than 9, it should be considered as 0.

After the first operand, the second operand can be entered using the same procedure. After both operands have been entered, the user can select the operator by pressing the momentary push button left (L) for addition, right (R) for subtraction, and down (D) for logically left shifting operand 1 by the count represented by operand 2.

The user can press up (U) to clear the seven segments and the stored numbers, and go back to entering the first operand.

Pushing up (U) twice in quick succession (within 1 second) toggles the calculator between the two operand mode described above or the accumulator mode. When in accumulator mode, the user only enters one operand and the other operand is the result of the previous calculation.

Message Board

Design a message board so a user can use a keyboard (using a [PS2 PMOD](#) or USB) to enter multiple messages. The entered messages should be scrolled across the seven segment displays one at a time with some pause between them. The right and left momentary push buttons should control the scroll rate (increasing or decreasing it). The up and down push momentary buttons should immediately start the previous or next message. Message orders are circular, “Next” message after last message is the first one, and “Previous” message of the first would be the last message. Note that the number of messages are not predefined, and depends on user input. The center momentary push button (C) should restart the scrolling of the message.

Video Controller

Design a VGA video controller that allows the user to control a VGA monitor using the BASYS 3 board. Then multiple multi-colored bitmap images should be displayed. The user can use left and right momentary push buttons to move to next and previous image. Image orders are circular, “Next” image after the last is the first one, and “Previous” image of the first would be the last one. The bitmaps should be loaded into a block memory.

RISC Processor

Design and implement a RISC processor. This project does not require the board, and can be tested in **post-implementation** simulation (this is not regular simulation). You do not need to implement interrupts.

The MIPS architecture is a good recommendation, however any RISC architecture is acceptable.

Your Own Design

You may come up with your own project, but it must be approved first. This requires that you submit a proposal of your project, which may be accepted or rejected with comments.