

# CSCE312 Project 4 In-Lab Exercises

Texas A&M University

March 27th, 2020

## 1 Instructions

1. You will be provided with 3 exercises to go through during lab.
  - (a) The first two exercises will be completed as a class with the respective TA's.
  - (b) If time permits, the last activity will be completed individually during the lab session. Students will have the chance to ask questions during lab for guidance, but will be expected to complete the exercise themselves.
2. Download the collateral from Slack and use the skeletal files provided. **Do not change the name of the files nor the names of the pins.**
3. Add your full name and UIN to the introductory comment present in each .asm file.
4. Implement the .asm file for each chip. Keep these chips for reference as you continue to work in assembly. These are *not for submission*.

## 2 Problem Set

1. **add.asm:** This program calculates the addition of two provided inputs, stored in RAM[0] (R0) and RAM[1] (R1). Compute the summation of the two inputs and store the result in R2.
  - (a) Input: R0 and R1 as the addends.
  - (b) Output: R2 as the summation.
2. **draw.asm:** This program takes keyboard inputs and draws to the screen based on the input. The conditions for what to draw are as follows:
  - (a) If KEY T is pressed, draw a horizontal line of 1 pixel width on the top row of the SCREEN.
  - (b) If KEY B is pressed, draw a horizontal line of 1 pixel width on the bottom row of the SCREEN.

- (c) Otherwise, clear the SCREEN.

Reminder from the Project 4 Handout: The screen is 512\*256 pixels with each RAM location pertaining to the display mapped to 16 pixels. It is 8192 (512/16 \* 256) RAM locations in total and you can use *@SCREEN* to set the A register to the start of the display. You write to the screen by setting bits inside that range. Refer to screen mapping discussion in lecture slides for details. The *@KBD* variable can be used to access the memory location that represents the keyboard. It will store the ASCII value of the key pressed or 0 if no key is pressed.

- 3. **pow.asm:** This program calculates the power of a number n raised to k, stored in RAM[0] (R0) and RAM[1] (R1) respectively. Compute the power of a number n raised to k and store the result in R2.

- (a) Input: R0 and R1 as the n and k respectively.
- (b) Output: R2 as the result.