

## Project 2: Basic Input/Output

This project is due on **October 18, 2019 at 6 p.m.** and counts for 5% of your course grade. Late submissions are not accepted. If you have a conflict due to travel, interviews, etc., please plan accordingly and turn in your project early.

The code and other answers you submit must be entirely your own work, and you are bound by the Honor Code. You may consult with other students about the conceptualization of the project and the meaning of the questions, but you may not look at any part of someone else's solution or collaborate with others to develop a solution. You may consult published references, provided that you appropriately cite them (e.g., with program comments), as you would in an academic paper.

Solutions must be submitted electronically via **Moodle**, following the submission checklist below.

---

### Introduction

In this project, you will program the DE10-Lite to perform basic input/output (I/O) using HEX displays, buttons, and/or LED strips. This project has 3 parts: **you are required to do only two of them.** You may choose any combination of two parts: For example, you may choose to do the scrolling part and buttons part. Alternatively, you can choose to do the buttons part along with the LED strip part. Note that the LED strip part requires some additional hardware (which can be purchased for around \$15 from Amazon).

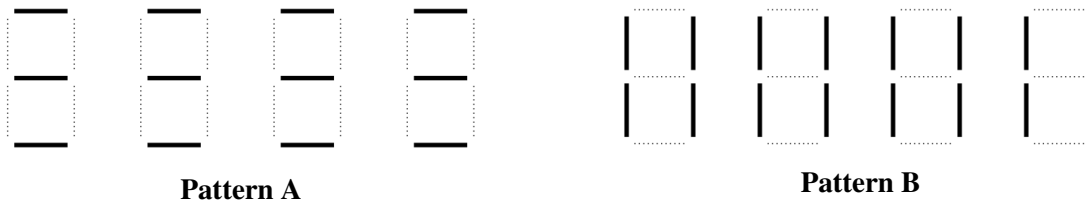
### Objectives:

- Understand how memory-mapped I/O (MMIO) is used to interact with peripherals.
- Understand memory alignment and how to access unaligned words.
- Appreciate the simplicity (and downsides!) of busy-wait loops for I/O and timing.

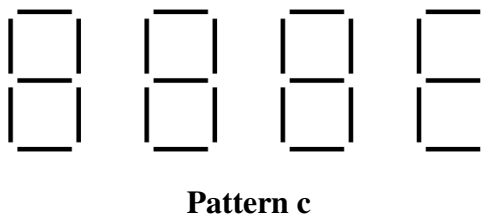
## Part 1. Scrolling Messages

Write a busy-wait assembly program that scrolls “Hello Buffs - - - \_ \_ \_ \_” across the HEX display of the DE10-lite board. (There are 4 spaces after the - - -, there are no spaces between the ‘-’). Only use the first 4 hex displays (the ones at 0xFF200020).

The program should then alternate between the following two patterns on the HEX display of the DE10-lite board:



These should be displayed 3 times, and then followed by the pattern (all 7 segments are lit):



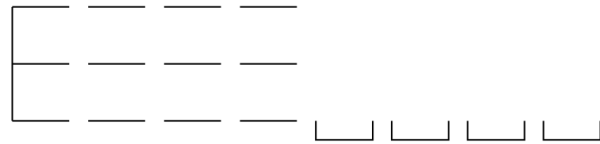
This pattern should be displayed 3 times alternating with a blank display. So, after the “Hello Buffs” message is scrolled across the display, the display should show the patterns: A, B, A, B, A, B, C, blank, C, blank, C, blank. The scrolling and display of patterns should repeat in an infinite loop.

The timing for the patterns and shifting can all be the same. Make it look good: the text should be readable, but also not crawl along too slowly.

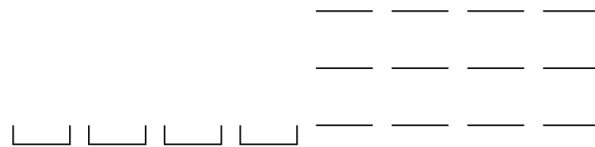
**What to submit** A `scroll.s` file that displays the previously described pattern.

## Part 2. Push Buttons

Write a second busy-wait assembly program that starts by scrolling the pattern:



from right to left across the HEX display. The \_ indicates a space (blank display). When either pushbutton 1 or 2 is pressed, the following pattern:



should start scrolling from left to right across the HEX display.

Each time either pushbutton 1 or pushbutton 2 is pressed, the display should change to the other mode of display (changing the direction of the scroll).

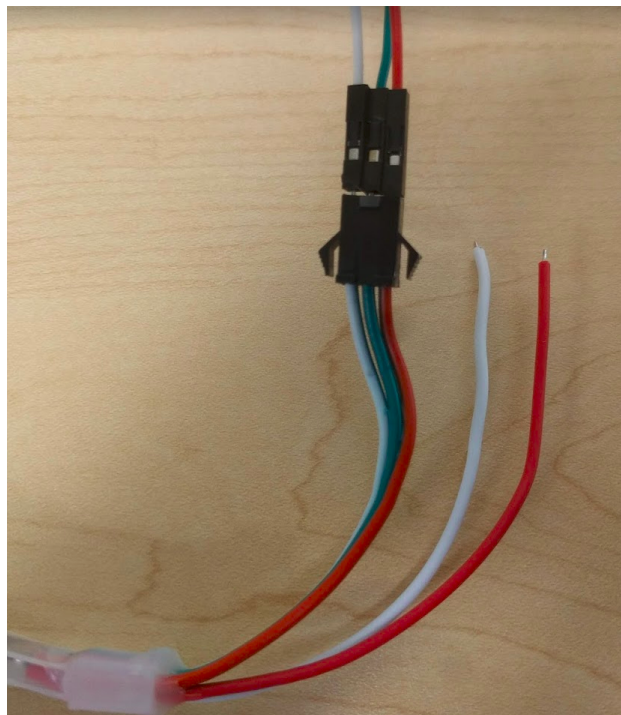
**What to submit** A `button.s` file that displays the described pattern.

## Part 3. LED strip

Write a busy-wait assembly program to control a WS2812 LED strip.

You can buy a WS2812-compatible LED strip from Amazon for \$10: [https://www.amazon.com/gp/product/B01LSF4QDM/ref=oh\\_aui\\_detailpage\\_o00\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B01LSF4QDM/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1). You are welcome to buy other LED strip variants as well, just ensure it is 5V and WS2812 (or SK6812)-compatible. More options (e.g. 144 LED per meter) can be purchased from Amazon, Adafruit, or Sparkfun.

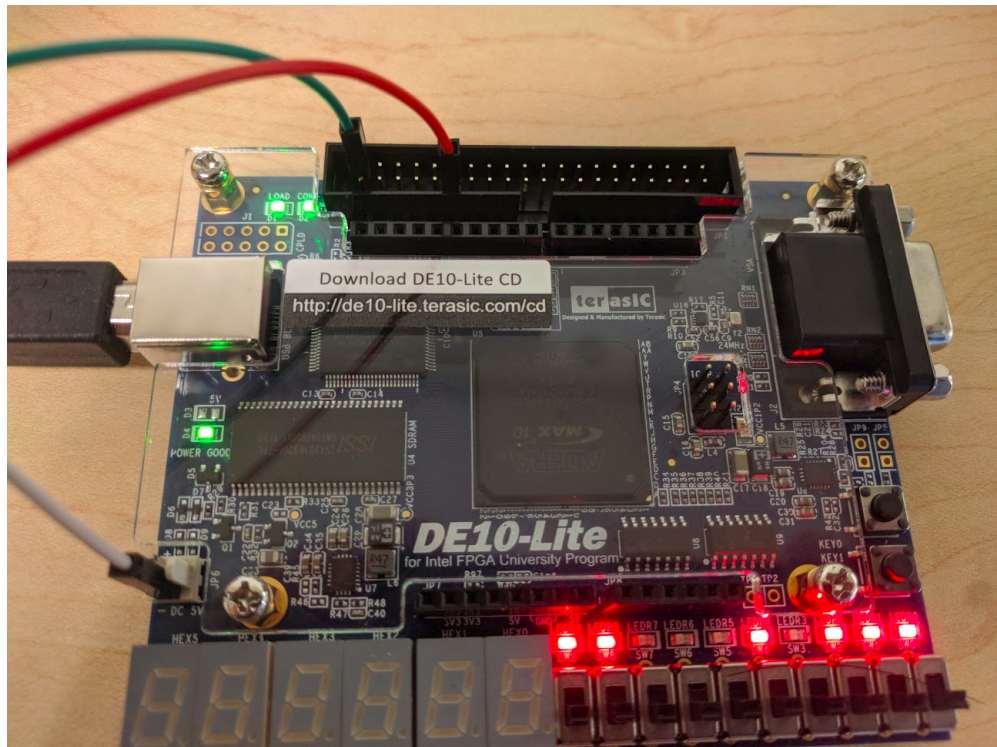
You'll also need 3 jumper connectors to connect your LED strip to the DE10-Lite. You can either purchase your own, borrow some from an embedded systems lab, or I have a few extras. You'll need three male-to-female connectors (you can also chain two male-male and female-female connectors). Ideally, you can match the colors of the LED strip (red, green, white), but this is not required.



Double check that your strand wires are white for ground (-), red for 5V (+), green for data. The red wire should be connected to +5V on your DE10 (JP1 header pin 11), and the white wire connected to ground. **It is important you do not mix these up**, as getting this backward may short out the LED strip and break it. I recommend connecting the white wire (ground) to the (-) (bottom) side of the DC 5V header to make it easy to double check. Make sure you connected ground to ground!

The green (data) wire should be connected to the JP1 header on pin 2 (upper left-most pin). See the DE10-Lite manual for a GPIO (JP1) pinout: [ftp://ftp.altera.com/up/pub/Intel\\_Material/Boards/DE10-Lite/DE10\\_Lite\\_User\\_Manual.pdf](ftp://ftp.altera.com/up/pub/Intel_Material/Boards/DE10-Lite/DE10_Lite_User_Manual.pdf)

Your connections should look like this:



**Ensure that the red (5V) and white (GND) are connected correctly**

Take a look at the WS2812 datasheet <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>, which describes how to drive the LEDs using carefully-timed signals. For example, to send a 0-bit, you must hold the data line high for 0.35 microseconds, and low for 0.8 microseconds. To send a 1-bit, hold the data line high for 0.7 microseconds and low for 0.6 microseconds. Setting the color of an LED takes 24-bits sent in this way, with 8-bits each for green, red, and blue values (0-255 brightness). The entire strand acts as a shift register, where writing a new 24-bit value to the first LED will shift its previous value to the next LED in the chain.

As a note/hint: The DE10-Lite computer has (approximately) a 50 MHz clock, and each instruction takes about 1 cycle to complete. Make sure you enable the *output* direction on whatever GPIO header pin you're using (see the DE10-Lite manual for direction bit MMIO address).

You will write **two** patterns for the LED strip. The first pattern will simply demonstrate that you can control the LED strip, and the second one is up to you!

### Part 3.1 Static Pattern

For the first pattern, you will create a static repeating pattern of red, green, and blue across the length of the LED strip:



Blue, red, green repeating pattern

Note: if you have trouble distinguishing between these colors (e.g. red-green color-blind), you may choose any 3 unique colors for your pattern.

## Part 3.2 Dynamic Pattern

For the second pattern, you will create any pattern that you like, as long as it changes over time. Here's a basic example: <https://youtu.be/hFDNiV1KmrC>.

Make it pretty! Imagine something you would bring to Burning Man or as a light show for Coachella.

### What to submit

1. A `led-static.s` file that displays a static pattern with different colors.
2. A `led-dynamic.s` file that displays a dynamic pattern of your choosing.

## Submission Checklist

Make sure you have chosen **two** parts:

1. Part 1: Scrolling Messages
2. Part 2: Push Buttons
3. Part 3: LED strip

Include all the files in their respective “what to submit” sections (e.g. `scroll.s`, `led-static.s`, `led-dynamic.s` if you did Part 1 and Part 3) in a gzipped tarball (`.tar.gz`) named `project2.your-identkey.tar.gz`.

You can make a tarball with `$ tar czf project2-erwu1234.tar.gz ./scroll.s ./led-static.s ./led-dynamic.s`.

Upload this tarball to **Moodle**.