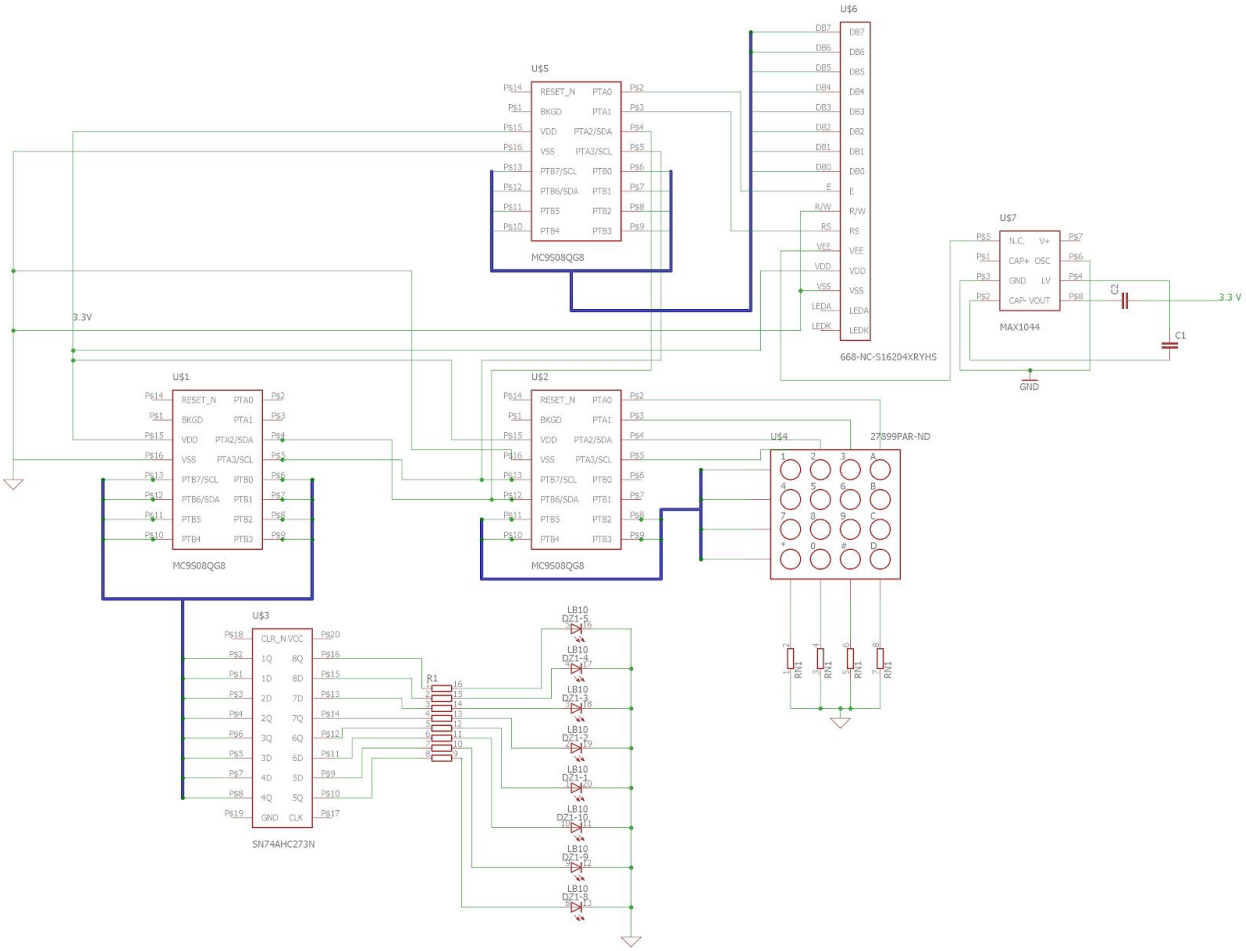
Memo to: Randy Larimer
From: Anthony Louis Rosenblum
Date: March 3rd, 2019
Regarding: Lab 2, LCD Character Display

Summary:

In this lab I designed a system in which sixten unique hex characters could be displayed on an LCD based on buttons pressed on a 4x4 keypad. One master HCS08 device read inputs from the keypad. This master then used I2C protocol to share the keypress data with two slave HCS08 devices. One slave displayed the relevant character on and LCD and the other slave displayed the binary value of the character on a set of LEDs. After filling the first row of the LCD the screen could be cleared with an accompanying command.

Circuit Schematic:



Master:

For the master device I had to implement a keyboard interrupt that would trigger whenever a key was pressed. I treated each row of the keypad as an output and alternated them one high at a time. When a button is pressed the row and the column are shorted and the microcontroller can detect the high while reading the column.

Because the rows and columns were each connected to four adjacent bits of PTBD I was able to store each keypress as the sum of the row value and the column value. This created the following lookup table:

$68 **(1)**	$64 **(2)**	$62 **(3)**	$61 **(A)**
$A8 **(4)**	$A4 **(5)**	$A2 **(6)**	$A1 **(B)**
$48 **(7)**	$44 **(8)**	$42 **(9)**	$41 **(C)**
$38 **(E)**	$34 **(0)**	$32 **(F)**	$31 **(D)**

Whenever a keypress triggered the interrupt these values would be sent on the data lines of the I2C bus. The LCD slave was addressed as $10 and the LED slave was addressed as $20.

LED Slave:

The LED slave would trigger its I2C interrupt whenever data came from the SDA line. Upon storing this data into a global variable a series of compare statements were used to determine the relevant data. The correct compare statement would then branch to a loop to display the corresponding binary information on a series of LEDs with 0 being 0000 and F being 1111.

LCD Slave:

The LCD slave had a significant amount of initialization to perform in order to get the LCD up and running. After providing power to and initializing the LCD the slave device then used to compare statements to determine what data was being received from the master. The relevant character would then be added onto the LCD and the cursor would shift one position to the right.

We chose to program the 7 key as the clear key. Upon pressing 7 the display would clear and then write the number 7 in the first position in the upper left.

Summary Comments:

This lab provided more experience with using the I2C module within the HCS08. It was also my first opportunity to work with an LCD. Despite the difficulty in getting the LCD to initialize and then also dealing with random unintelligible characters displaying I enjoyed being able to experience how LCD technology is operated.