CSE 379

University at Buffalo

Keming Kuang (kemingku)

Eric Li (eli9)

March 15, 2018

Contents

1 Division of work

There was no division of work for this lab. This whole lab was completed together as a collaboration between Eric and Keming. We both gave inputs on how to tackle the four subroutines before deciding the approach. We both committed equally towards the completion of this lab.

2 Description

2.1 Purpose of the program:

The objective of this whole program and its subroutines is to learn how to use the GPIO. Throughout the whole lab the most challenging aspect was learning how to use the GPIO.

2.2 Description:

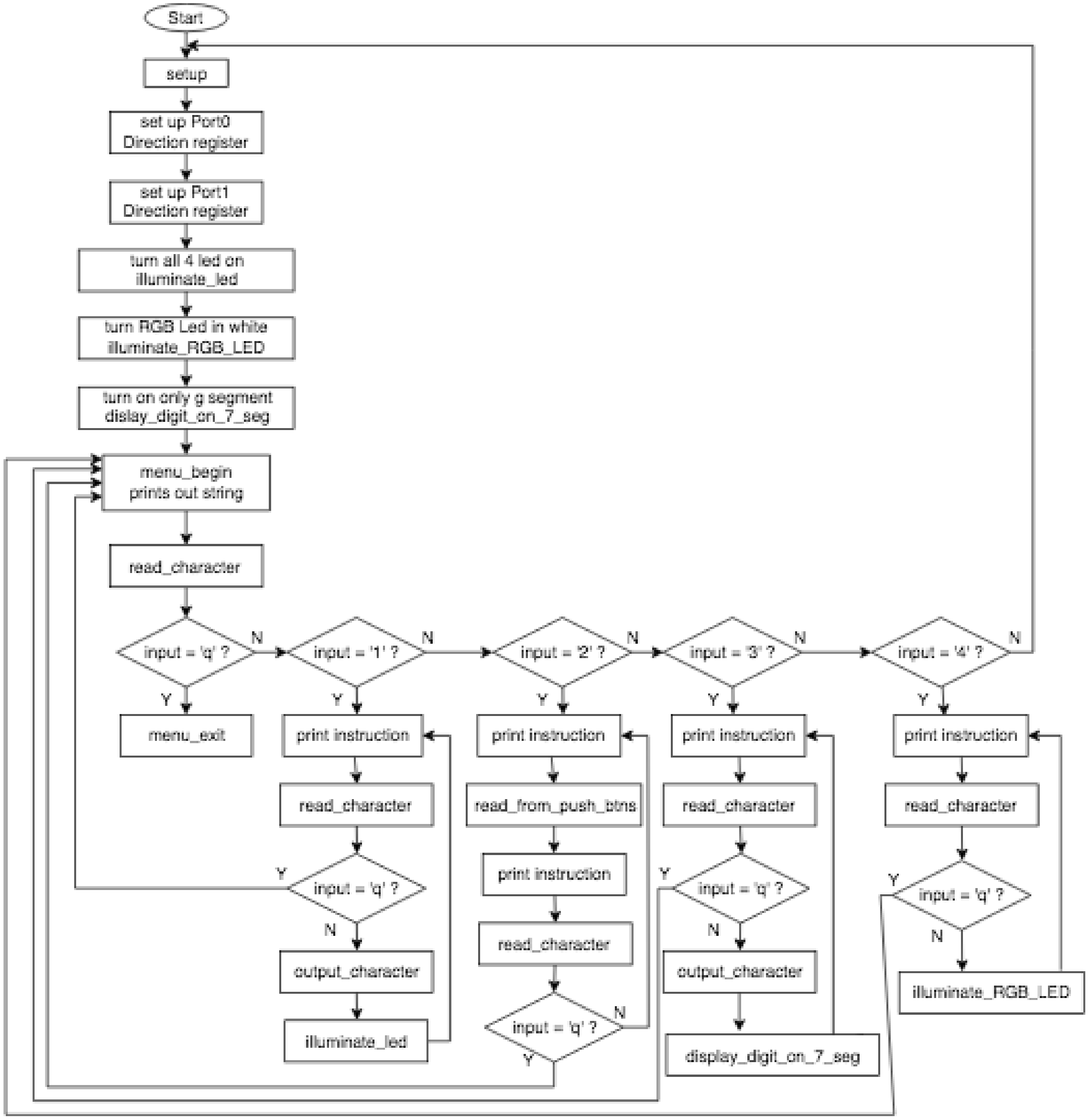
This program has four functionalities, illuminate LED, read from momentary push buttons, display digit on seven segment display, and illuminate RGB LED. The main program provides instruction and sets up the user interface to run these four functionalities. Illuminate LED reads a number from 0-15 and lights up the LED to match the numbers binary value. Read from momentary push buttons, reads the binary value conveyed from the momentary push buttons and prints the value in decimal to putty. Display digit on seven segment display reads in a hexadecimal number from 0-F and displays it on the seven-segment display. Illuminate RGB LED reads in a number from 1-7 and switches the color of the RGB LED.

2.3 Debugging steps

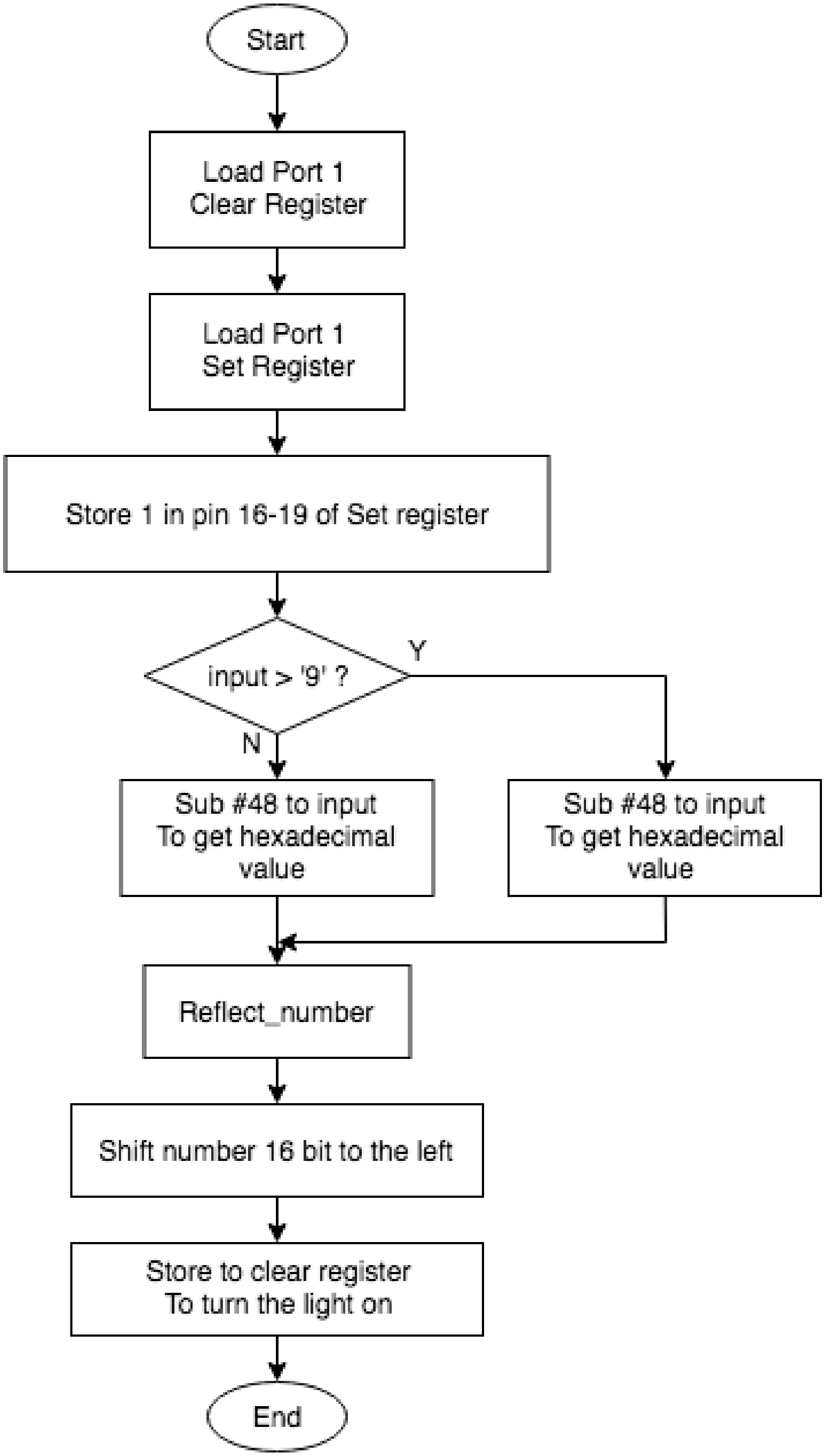
To debug the program simply run it and instructions will be printed on putty. When typing entering A-F make sure it is uppercase.

3 Flow Chart

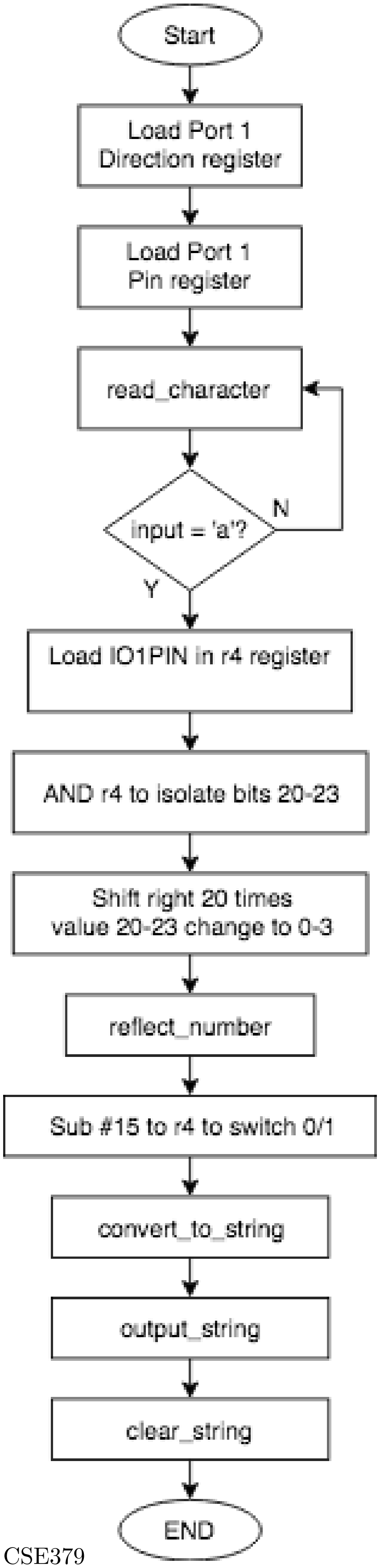
3.1 lab4.s Main code



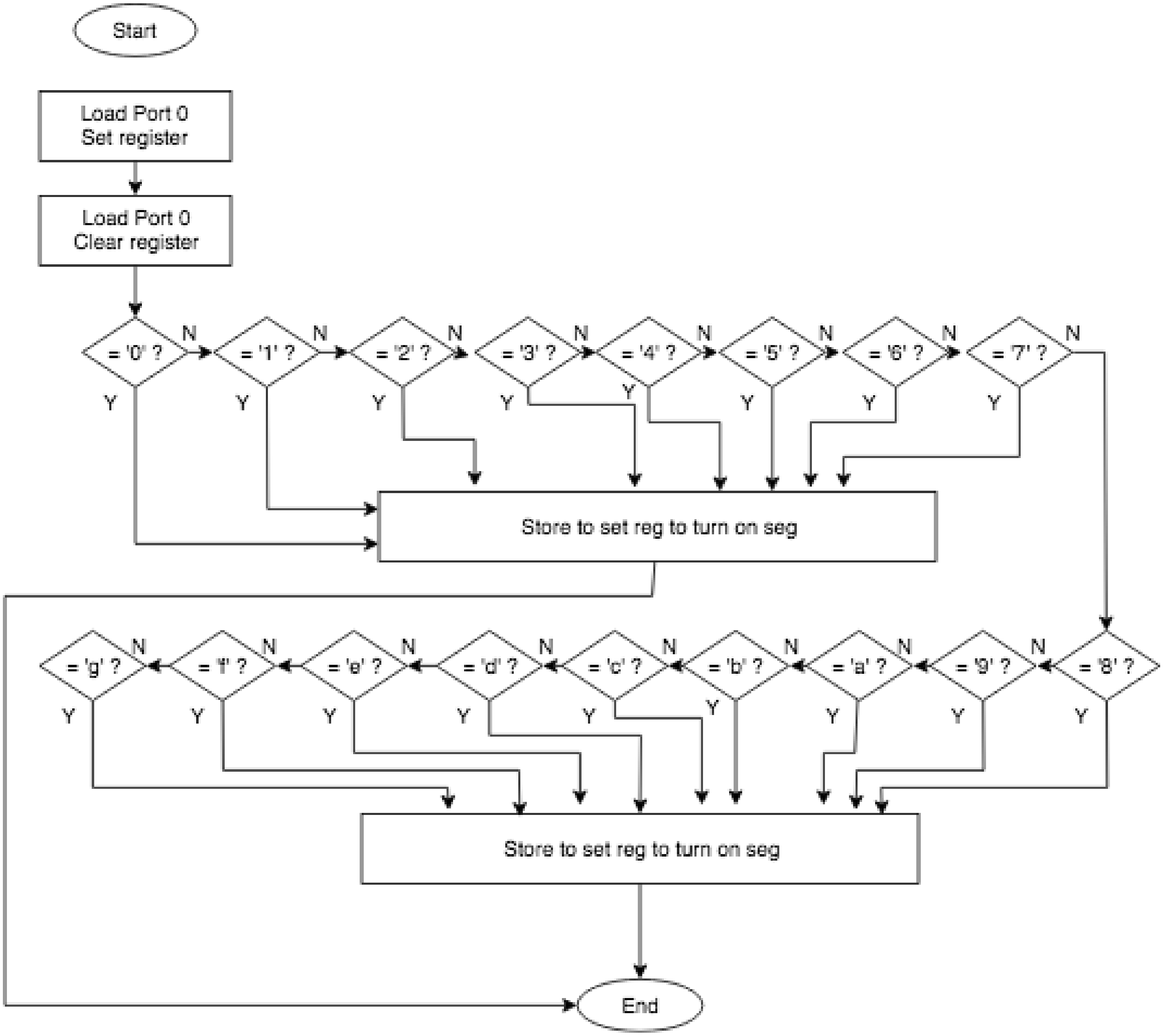
3.2 illuminate_led



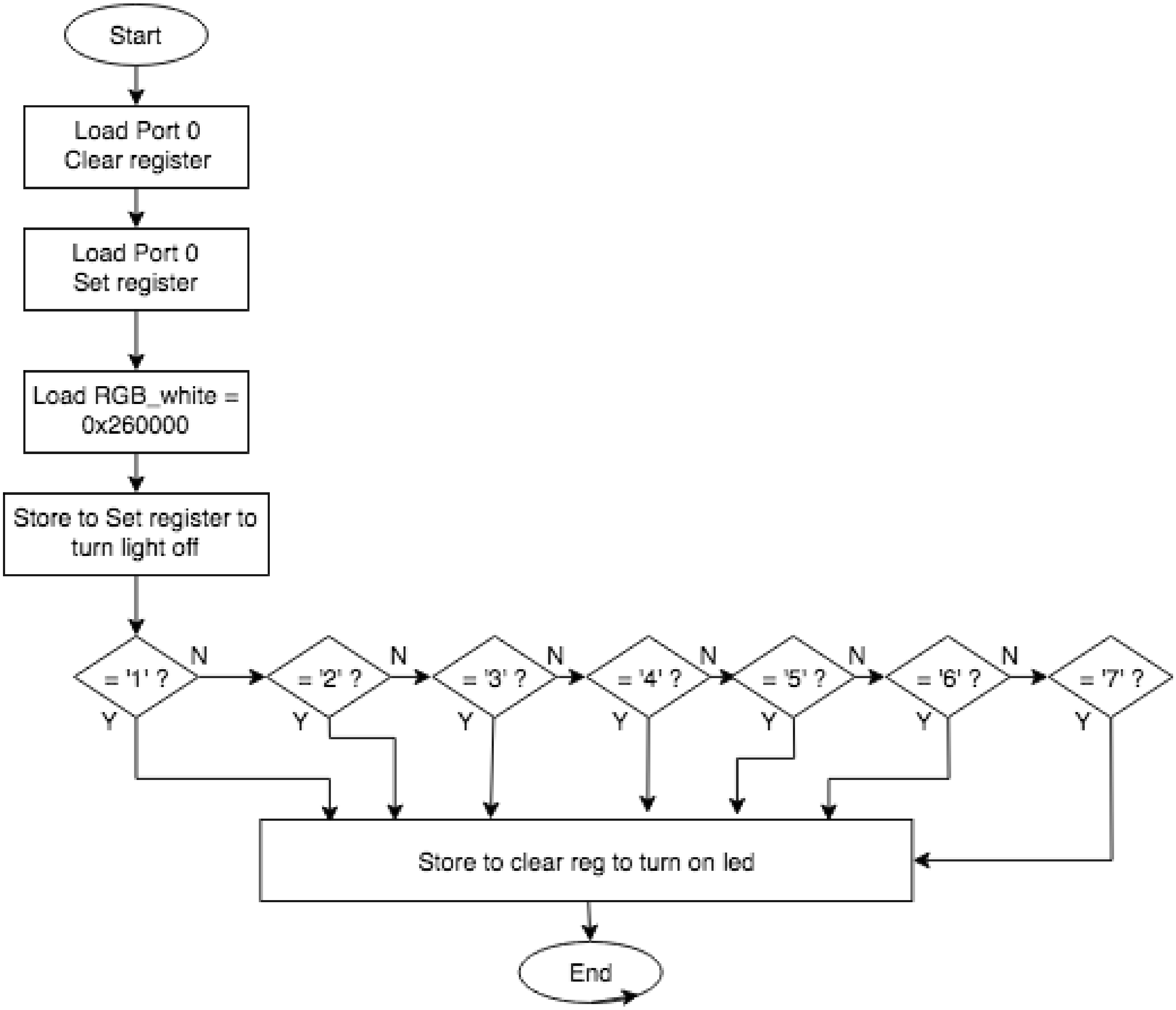
3.3 read_from_push_btns



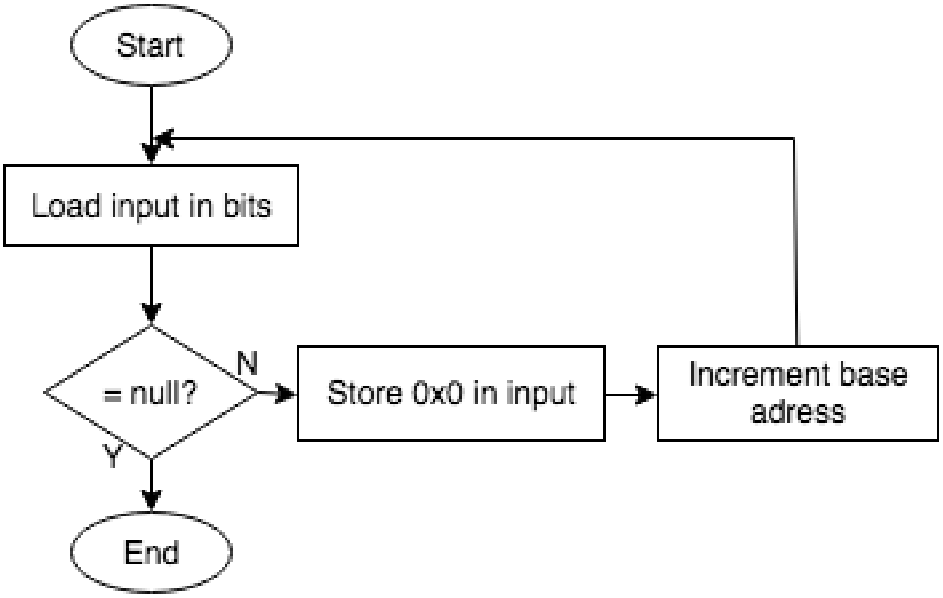
3.4 display_digit_on_7_seg



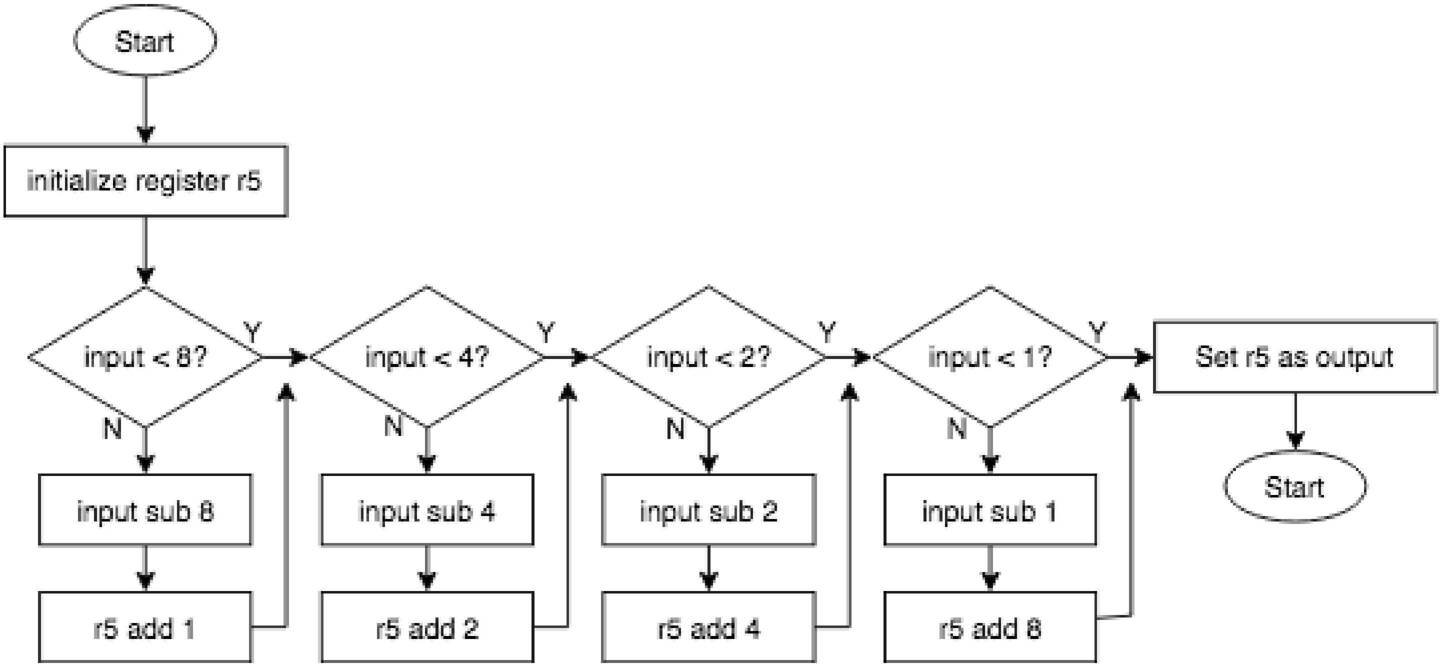
3.5 illuminate_RGB_LED



3.6 clear_string



3.7 reflect_number



4 Summary

4.1 lab4.s Main code:

The purpose of the main program lab4.s is mainly to set up the user interface and to initiate the subroutines. The first thing the main program does is to run the setup subroutine which sets up IO0DIR and IO1DIR and prepares the initial state. The Main program will then read in a character if the character is 1-4 it will branch to the respective subroutines associated with numbers 1-4. If the read character is q then the program terminates, this only happens in the main menu, otherwise if q is read the program will branch back to the main menu. The subroutines associated with numbers 1-4 branches to illuminate led and etcetera only if the read character matches certain conditions outlined by the instructions.

4.2 Illuminate LED:

The logic of illuminate LED is relative simple create a hexadecimal value to store into IO1CLR. This is accomplished by first reading in a character then converting that number to a decimal value. After converting the number to a decimal value, this number must be reflected because the MSB is the 16th bit rather than the 19th bit. After the number is reflected it is then shifted left 16 bits before being stored in IO1CLR.

4.3 Read from momentary push buttons:

The logic for this subroutine is that after a certain key press IO1PIN will be read. After IO1PIN is read the value is shifted 20 bits to the right to make the 20th to 23rd bit the 0th and 3rd bit. This value is then reflected before having its 0th to 3rd bits flipped, 1s becomes 0s and 0s becomes 1s. This new value is then converted into a string before it is printed to putty.

4.4 Display digit on seven segment display:

The logic behind this subroutine is very simple. This subroutine is a bunch of comparison statements one for each number 0-15. There is a library of hexadecimal values which turns on the seven-segment display for each number from 0 to 15. If 5 is read, it will be compared to every number 0 - 15. Once it reaches the comparison for 5 the hexadecimal value for the seven-segment display to display 5 is stored into IO0SET.

4.5 Illuminate RGB LED:

This subroutine is a set of comparison statements which compares a read number to 1-7. If read number is equal to any number 1-7 it will take the hexadecimal value of the color that number represents and stores it in IO0CLR.

4.6 Clear string:

This subroutine takes a string and clears the string. This is done by pulling a character from the string comparing it to null if it is not null it will be replaced with null. The string will then be incremented, this process repeats until a null character is read.

4.7 Reflect number:

This subroutine takes in a number in binary and reflects its bits. This is done with a bunch of comparison statements. For this lab this routine can only reflect numbers represented by 4 bits. The number is first compared to the number 8, if the number is greater, 1 is added to the result and 8 is subtracted from the number being reflected. Next the number is compared to 4, if the number is greater, 2 is added to the result and 4 is subtracted from the numbering being reflected. This process repeats until the final comparison to 1.