CSE 379-Intro to Microprocessors

Lab #4 - Lab Report

Henry Dacres(henrydac) & Eric Liu(eliu4)

Division of Work	
Henry Dacres	Eric Liu
● Quality Control ● Illuminate_LEDs ● Illuminate_RGB ● Flowsheet	● Documentation ● Linking code various subroutines ● Debugging ● Subroutine Summaries

Purpose of Program

In this lab, students will use I/O to interface hardware with the ARM processor. During the lab, we created a library for holding our subroutines. For this lab, we wrote four ARM assembly subroutines. Titled read_from_push_btns, illuminate_LEDs, illuminate_RGB_LED, and read_from_keypad. After the completion and testing of these subroutines, another subroutine called lab4 was written this subroutine starts by presenting a menu on the terminal that allows you to repeatedly Illuminate the LED, Illuminate the RGB components, read user input from the Push Buttons or read user input from the keypad.

The Program will display a binary value between 0000 and 1111 on the LEDs. LED3 is the most significant bit while the LED0 is the LSB. Pressing a 1 will turn on the LED and a 0 should turn it off. The program will read a value from the momentary push buttons. The binary value read will be displayed in hexadecimal in PuTTy. The key that is pressed on the keypad, will display in PuTTy. When illuminating the RGB LED, the user is allowed to select the color (red, green, blue, purple, yellow, or white) via PuTTy.**While using the program please refrain from using the backspace character**

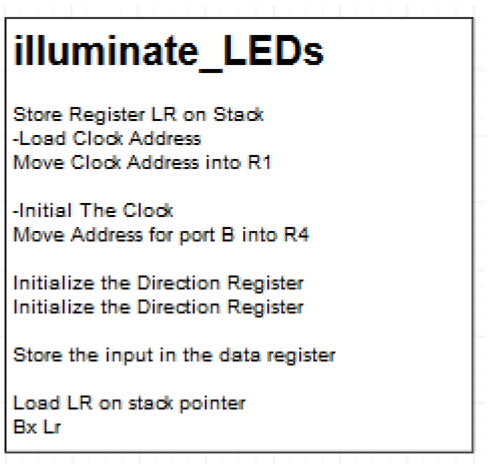
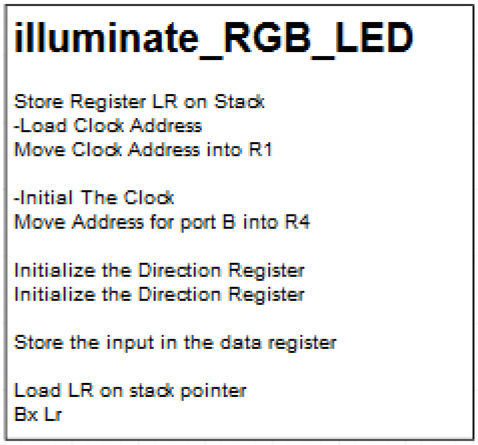
University at Buffalo The State University of New York

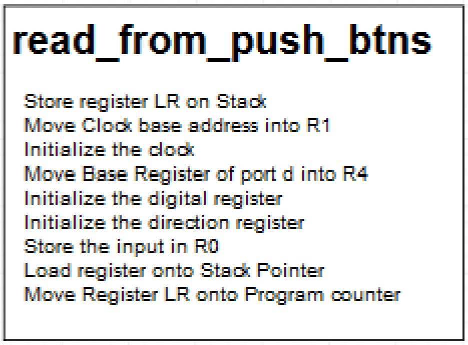
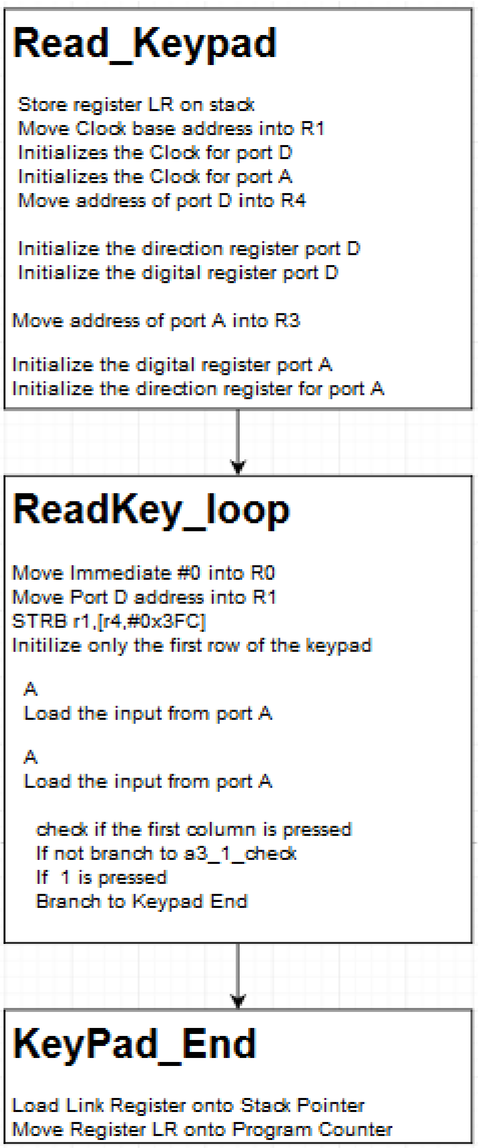
Logic

Subroutine	**Summary**
uart_init	Initializes the user UART for use.
output_character	Transmits a character passed into the routine in r0 to PuTTy via the UART
read_character	Reads a character from PuTTy via the UART, and returns the character in r0.
read_string	A string entered in PuTTy and stores it as a null-terminated string in memory.
output_string	Displays a null-terminated string in PuTTy. The base address of the string should be passed into the routine in r4
read_from_push_btns	Reads the momentary push buttons, and returns the value read in r0. Push button 2 should correspond to the MSB and 5 to the LSB
Illuminate_LEDs	Illuminates the four LEDs. The pattern indicating which LEDs to illuminate is passed into the routine in r0. Bit 3 corresponds to LED 3, bit 2 to LED 2, bit 1 to LED 1, and bit 0 to LED0
illuminate_RGB_LED	Illuminates the RGB LED. The color to be displayed is passed into the routine in r0.
read_from_keypad	Reads a keypress on the keypad, and returns the value corresponding to the key that was pressed in r0.
lab4loop	Prints out the Menu and prompts user

	to enter user input
lab4led	Output the prompts for LED, reads user input, will illuminate the LED.
lab4rgb:	outputs the prompt for rgb outputs the second prompt for rgb reads user input converts the input to int Outputs
lab4pushbut	Output the pushbutton prompt reads user input.
lab4keypad	Reads keypad for user input routine
singledig	If the value passed in is a single digit, it will convert it to a number.
pushbutoutput	Outputs second prompt,outputs answer to screen

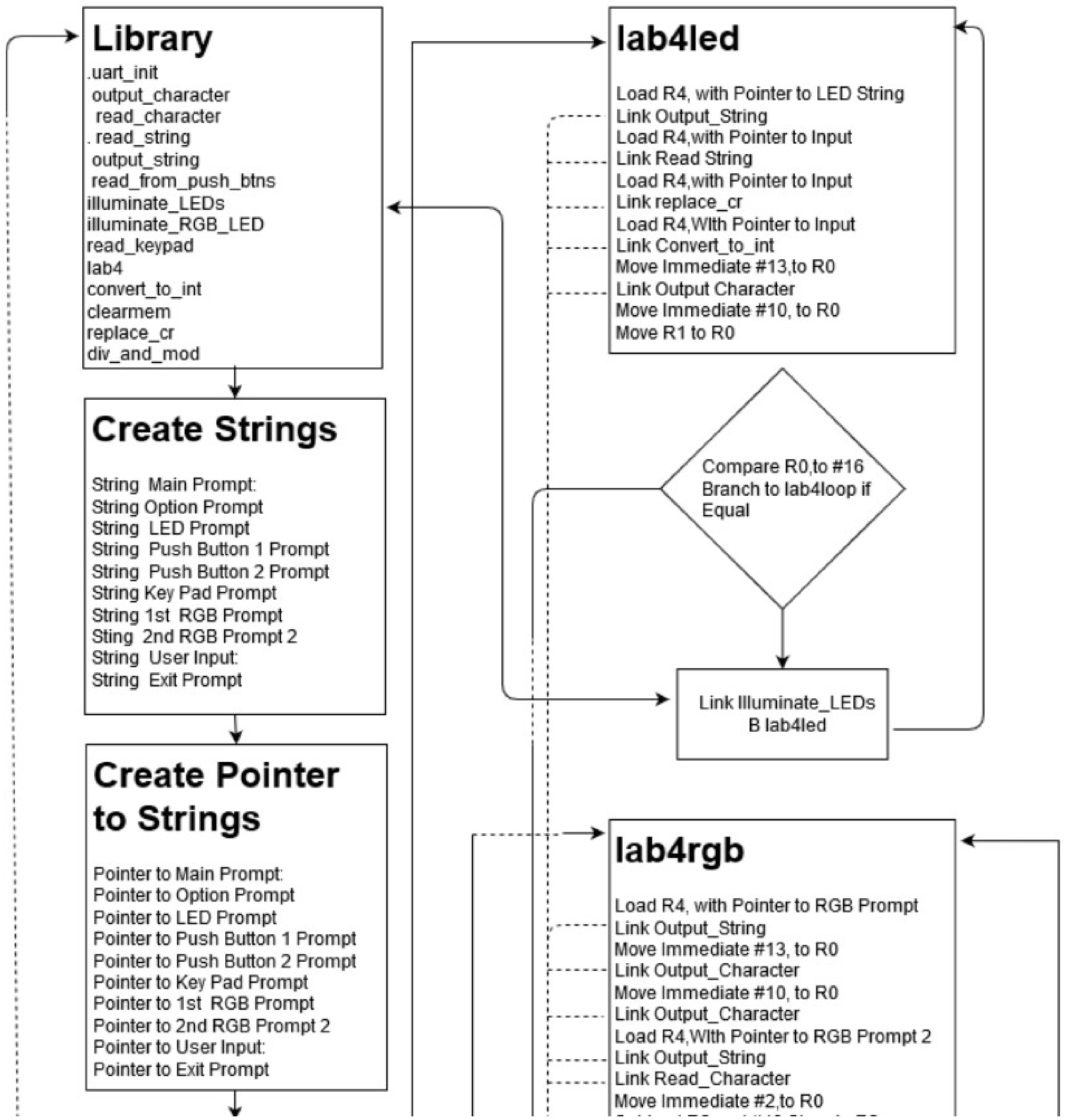
Flow Charts

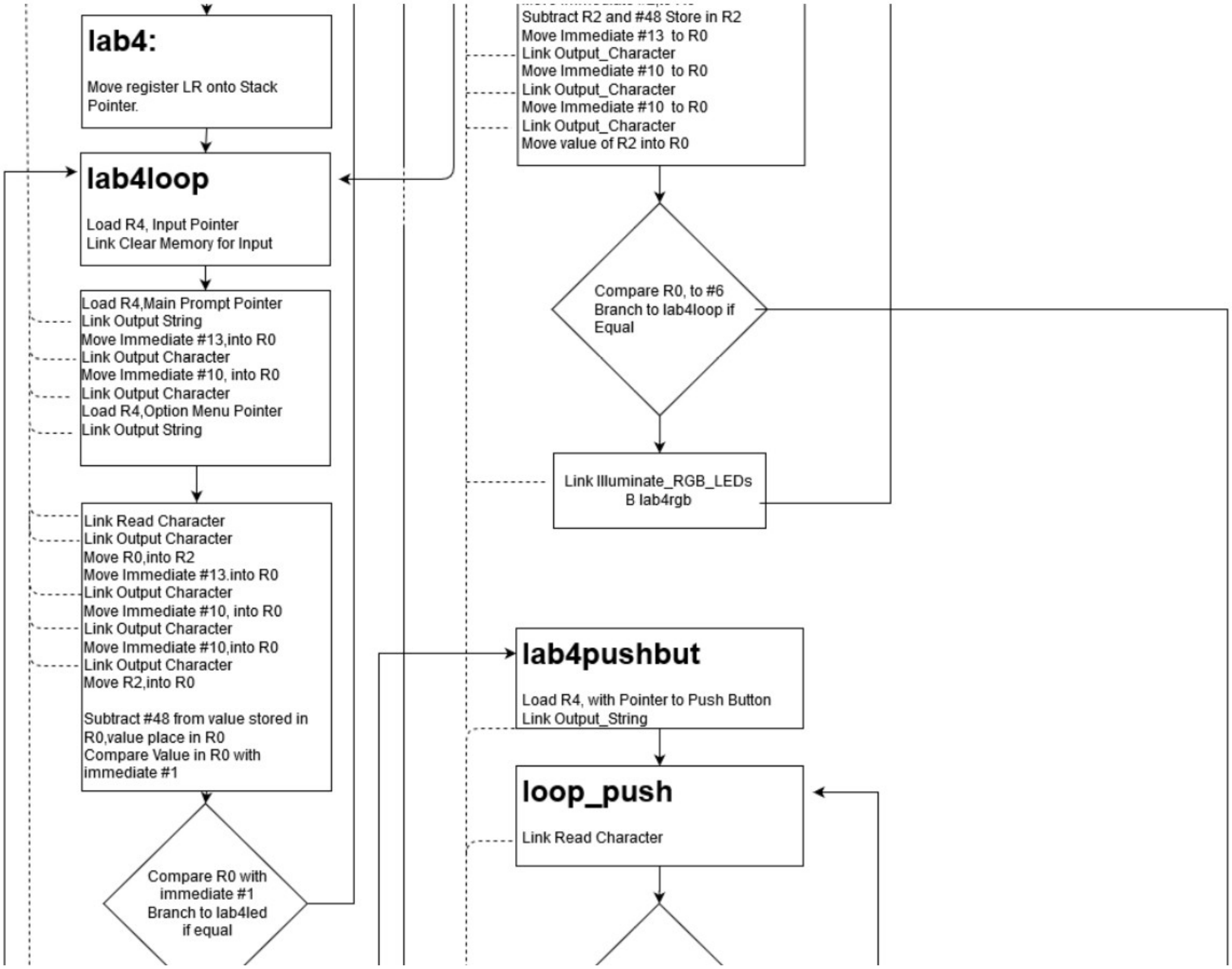
Illuminate_LED	 illuminate_LEDs Store Register LR on Stack -Load Clock Address Move Clock Address into R1 -Initial The Clock Move Address for port B into R4 Initialize the Direction Register Initialize the Direction Register Store the input in the data register Load LR on stack pointer Bx Lr
Illuminate_RGB_LED	 illuminate_RGB_LED Store Register LR on Stack -Load Clock Address Move Clock Address into R1 -Initial The Clock Move Address for port B into R4 Initialize the Direction Register Initialize the Direction Register Store the input in the data register Load LR on stack pointer Bx Lr

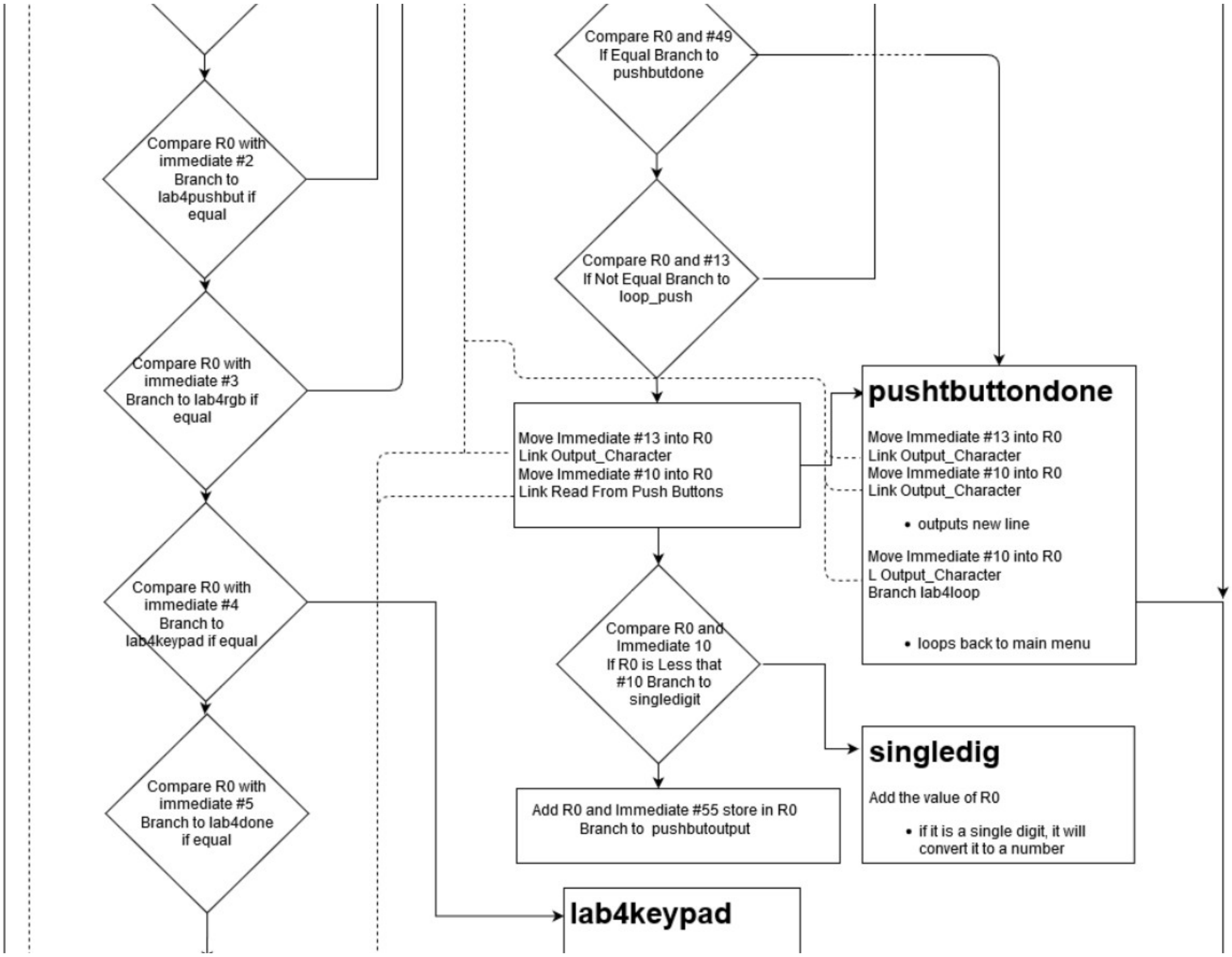
read_from_push_btn	 read_from_push_btns Store register LR on Stack Move Clock base address into R1 Initialize the clock Move Base Register of port d into R4 Initialize the digital register Initialize the direction register Store the input in R0 Load register onto Stack Pointer Move Register LR onto Program counter
Read_KeyPad	 Read_Keypad Store register LR on stack Move Clock base address into R1 Initializes the Clock for port D Initializes the Clock for port A Move address of port D into R4 Initialize the direction register port D Initialize the digital register port D Move address of port A into R3 Initialize the digital register port A Initialize the direction register for port A ReadKey_loop Move Immediate #0 into R0 Move Port D address into R1 STRB r1,[r4,#0x3FC] Initilize only the first row of the keypad A Load the input from port A A Load the input from port A check if the first column is pressed If not branch to a3_1_check If 1 is pressed Branch to Keypad End KeyPad_End Load Link Register onto Stack Pointer Move Register LR onto Program Counter

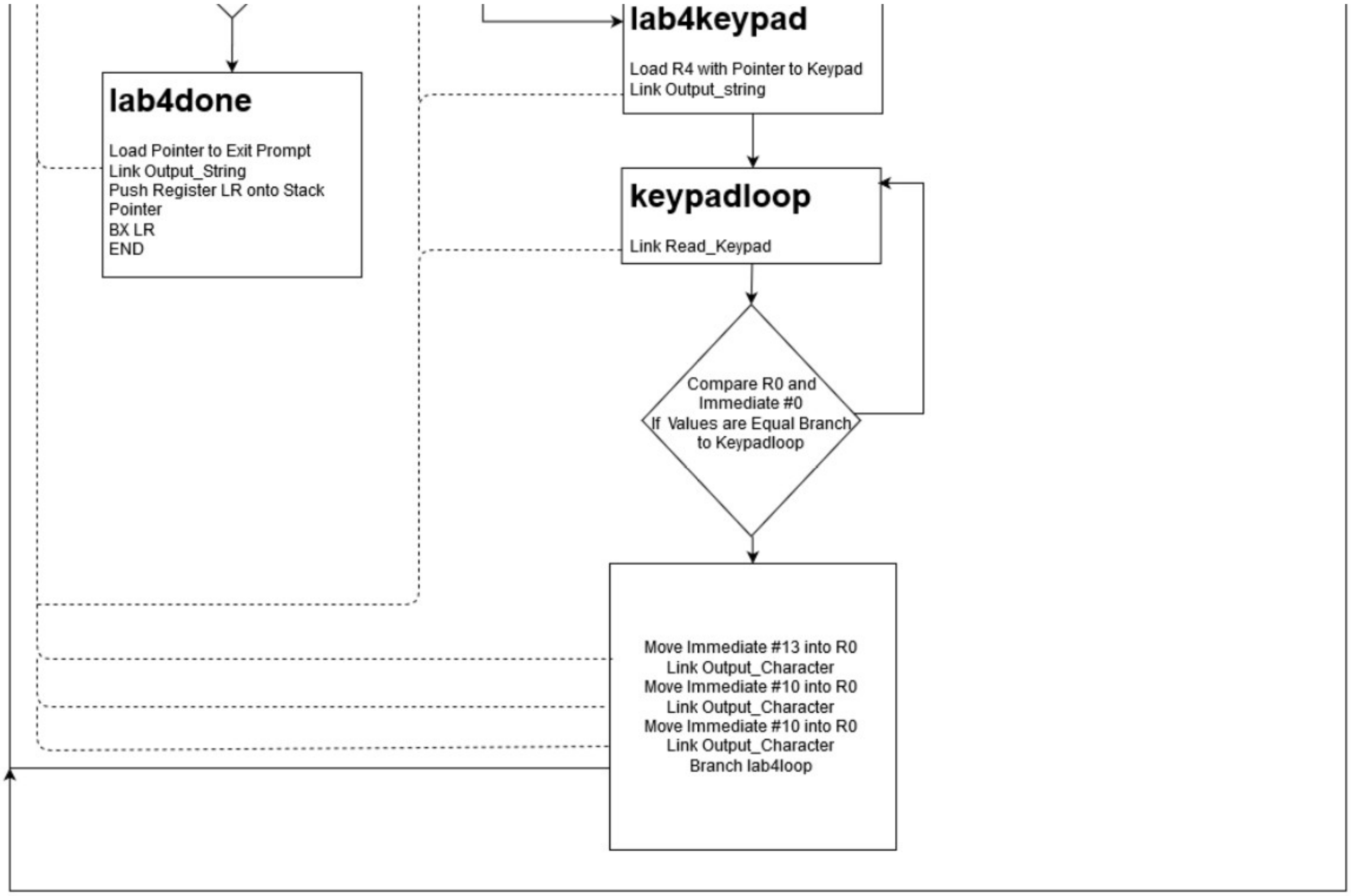
Lab 4 Flowsheet

https://drive.google.com/file/d/1H7drDIt8S2RaSJLtW86iTar6BXxqVBXC/view?usp=sharing




lab4:
Move register LR onto Stack Pointer.
lab4loop
Load R4, Input Pointer
Link Clear Memory for Input
Load R4,Main Prompt Pointer
Link Output String
Move Immediate #13,into R0
Link Output Character
Move Immediate #10, into R0
Link Output Character
Load R4,Option Menu Pointer
Link Output String
Link Read Character
Link Output Character
Move R0,into R2
Move Immediate #13.into R0
Link Output Character
Move Immediate #10, into R0
Link Output Character
Move Immediate #10,into R0
Link Output Character
Move R2,into R0
Subtract #48 from value stored in R0,value place in R0
Compare Value in R0 with immediate #1
Compare R0 with immediate #1
Branch to lab4led
if equal
Subtract R2 and #48 Store in R2
Move Immediate #13 to R0
Link Output_Character
Move Immediate #10 to R0
Link Output_Character
Move Immediate #10 to R0
Link Output_Character
Move value of R2 into R0
Compare R0, to #6
Branch to lab4loop if Equal
Link Illuminate_RGB_LEDs
B lab4rgb
lab4pushbut
Load R4, with Pointer to Push Button
Link Output_String
loop_push
Link Read Character


Compare R0 with immediate #2 Branch to lab4pushbut if equal
Compare R0 with immediate #3 Branch to lab4rgb if equal
Compare R0 with immediate #4 Branch to lab4keypad if equal
Compare R0 with immediate #5 Branch to lab4done if equal
Compare R0 and #49 If Equal Branch to pushbutdone
Compare R0 and #13 If Not Equal Branch to loop_push
Move Immediate #13 into R0
Link Output_Character
Move Immediate #10 into R0
Link Read From Push Buttons
Compare R0 and Immediate 10 If R0 is Less that #10 Branch to singledigit
Add R0 and Immediate #55 store in R0
Branch to pushbutoutput
pushtbuttondone
Move Immediate #13 into R0
Link Output_Character
Move Immediate #10 into R0
Link Output_Character
• outputs new line
Move Immediate #10 into R0
L Output_Character
Branch lab4loop
• loops back to main menu
singledig
Add the value of R0
• if it is a single digit, it will convert it to a number
lab4keypad


lab4done
Load Pointer to Exit Prompt
Link Output_String
Push Register LR onto Stack Pointer
BX LR
END
lab4keypad
Load R4 with Pointer to Keypad
Link Output_string
keypadloop
Link Read_Keypad
Compare R0 and Immediate #0
If Values are Equal Branch to Keypadloop
Move Immediate #13 into R0
Link Output_Character
Move Immediate #10 into R0
Link Output_Character
Move Immediate #10 into R0
Link Output_Character
Branch lab4loop