CSE 379

University at Buffalo

Keming Kuang (kemingku)

Eric Li (eli9)

March 27, 2018

Contents

1 Division of work

This lab was a team effort between Eric Li and Keming Kuang. The entirety of this lab was written and thought through together as a team. The logic was thought through together, the code was written together, lastly learning the intricacies of using interrupts was also done as a team. What is presented is the effort of both Eric Li and Keming Kuang.

2 Description

2.1 Objective:

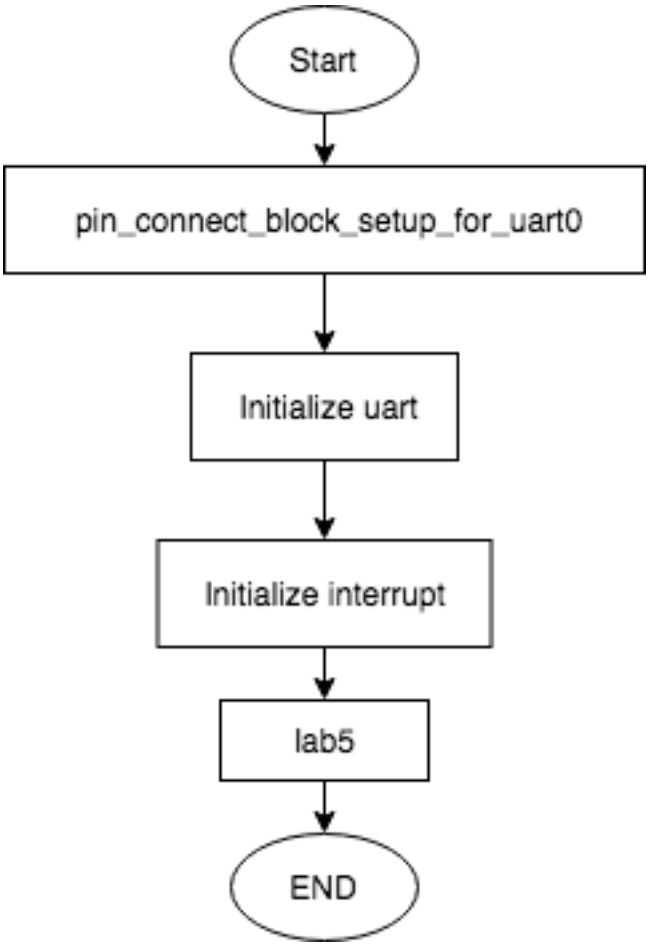
The objective of this lab assignment is to learn firstly, what are interrupts, the types of interrupts, how to set them up, and lastly how to use them in practice. The culmination is to use this knowledge to code a program which, will turn on/off the seven-segment display when push button p014 is pressed on the ARM board, and to display a hexadecimal number entered through putty on the seven-segment display. The program also saves the previous state and ignores putty inputs when the seven-segment display is off.

2.2 Debugging steps

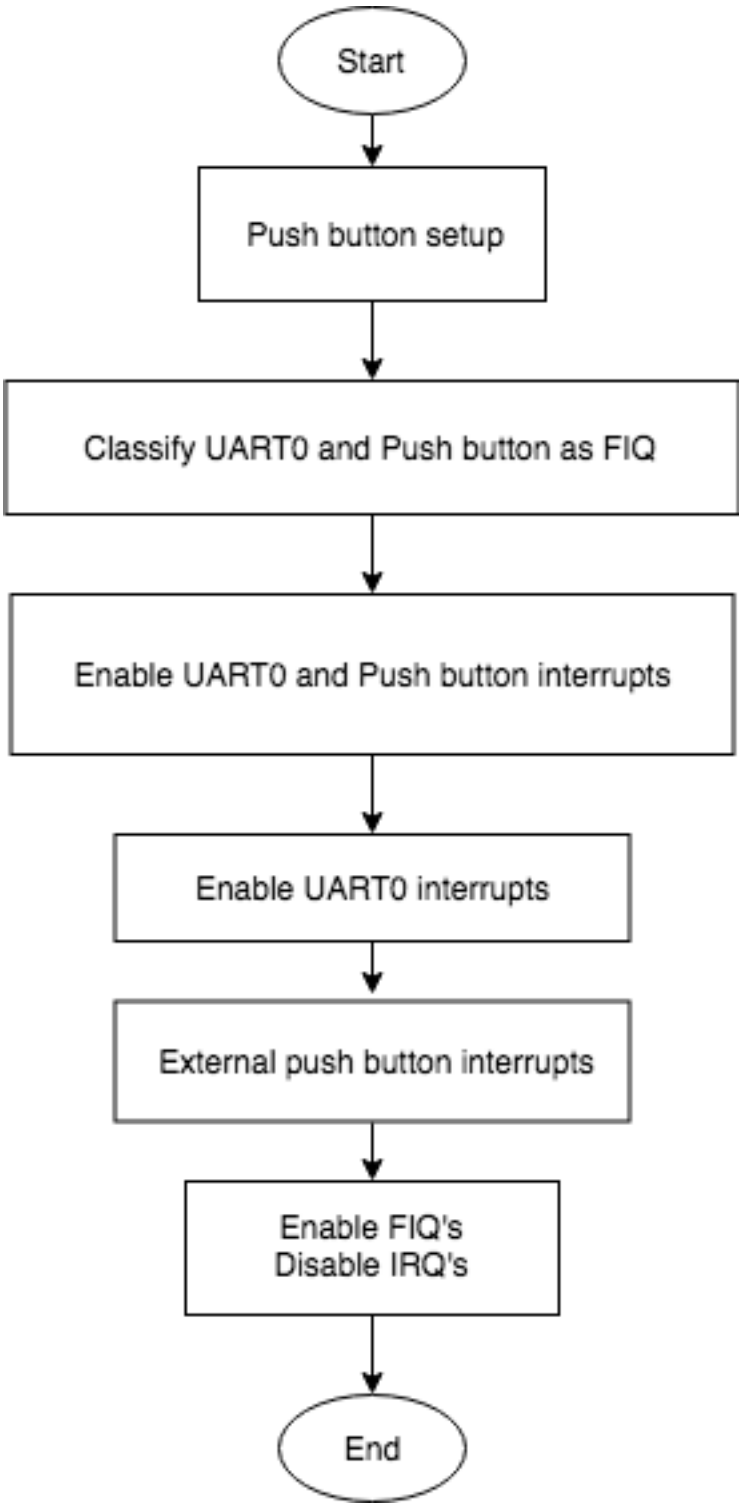
To debug and test the program simply run it and load it to the ARM board. Instructions will then be displayed via putty when. The seven-segment display should initially be in a off state. Turning on the display will light up the g segment of the display. Entering hexadecimal inputs via putty will display the number on the display. Pressing the push button will turn the display on/off and restores the display to its previous state when turned on. When the display is off, all inputs via putty will be ignored. To terminate the program simply enter q via putty. This will also turn off the display if it is currently on when q is entered.

3 Flow Chart

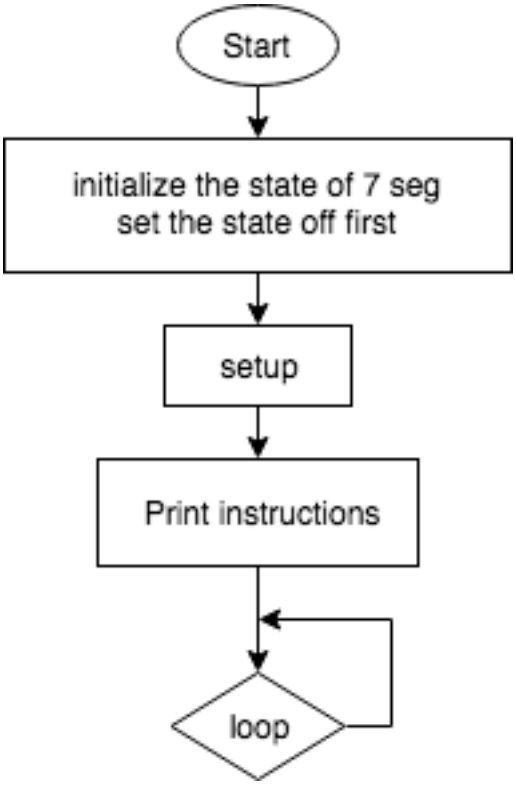
3.1 lab5 wrapper Main code



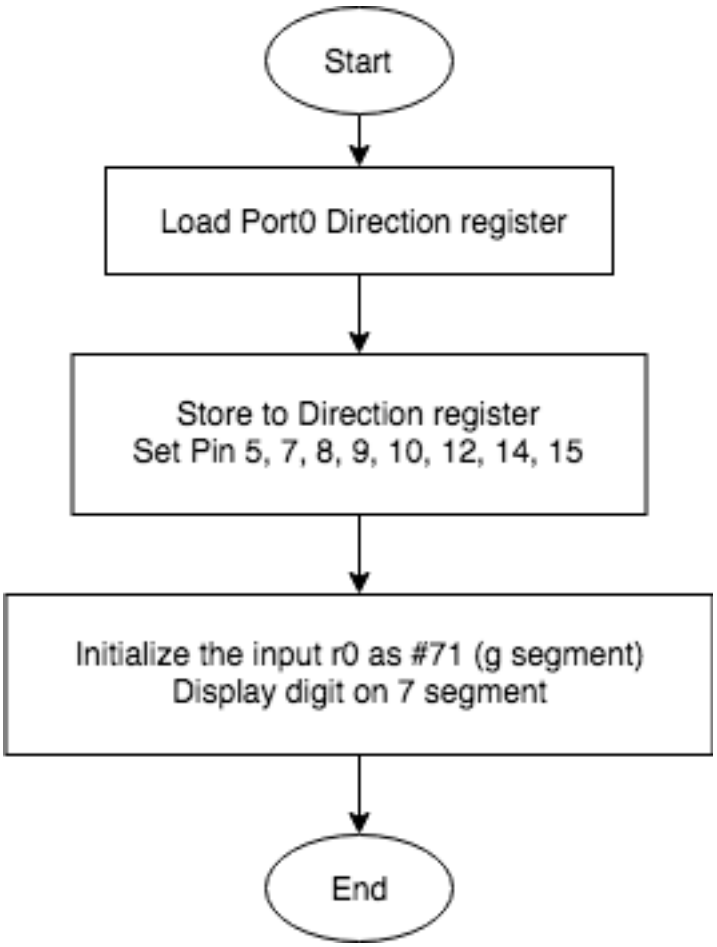
3.2 initialize interrupt



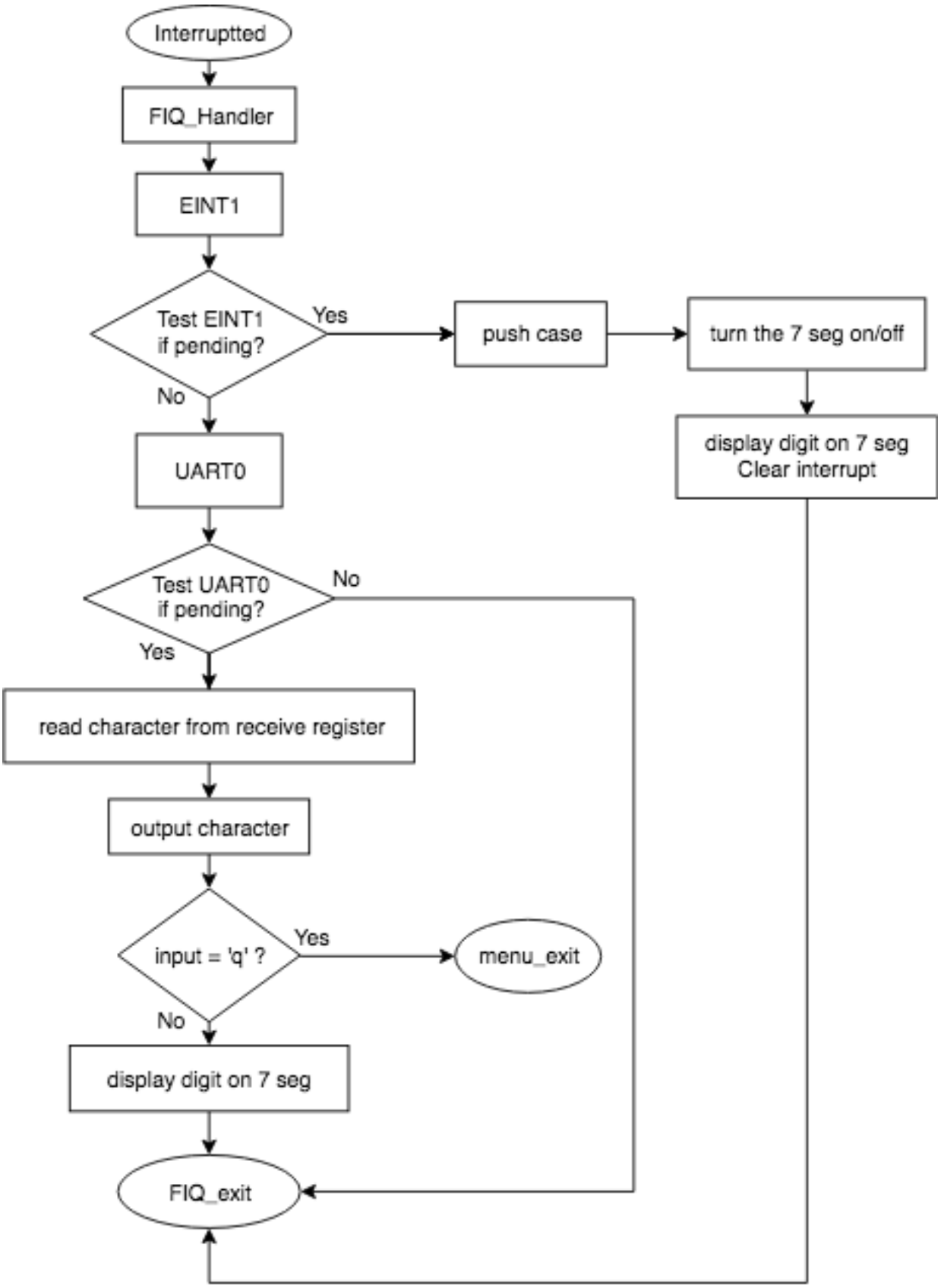
3.3 Maincode


Start
initialize the state of 7 seg
set the state off first
setup
Print instructions
loop

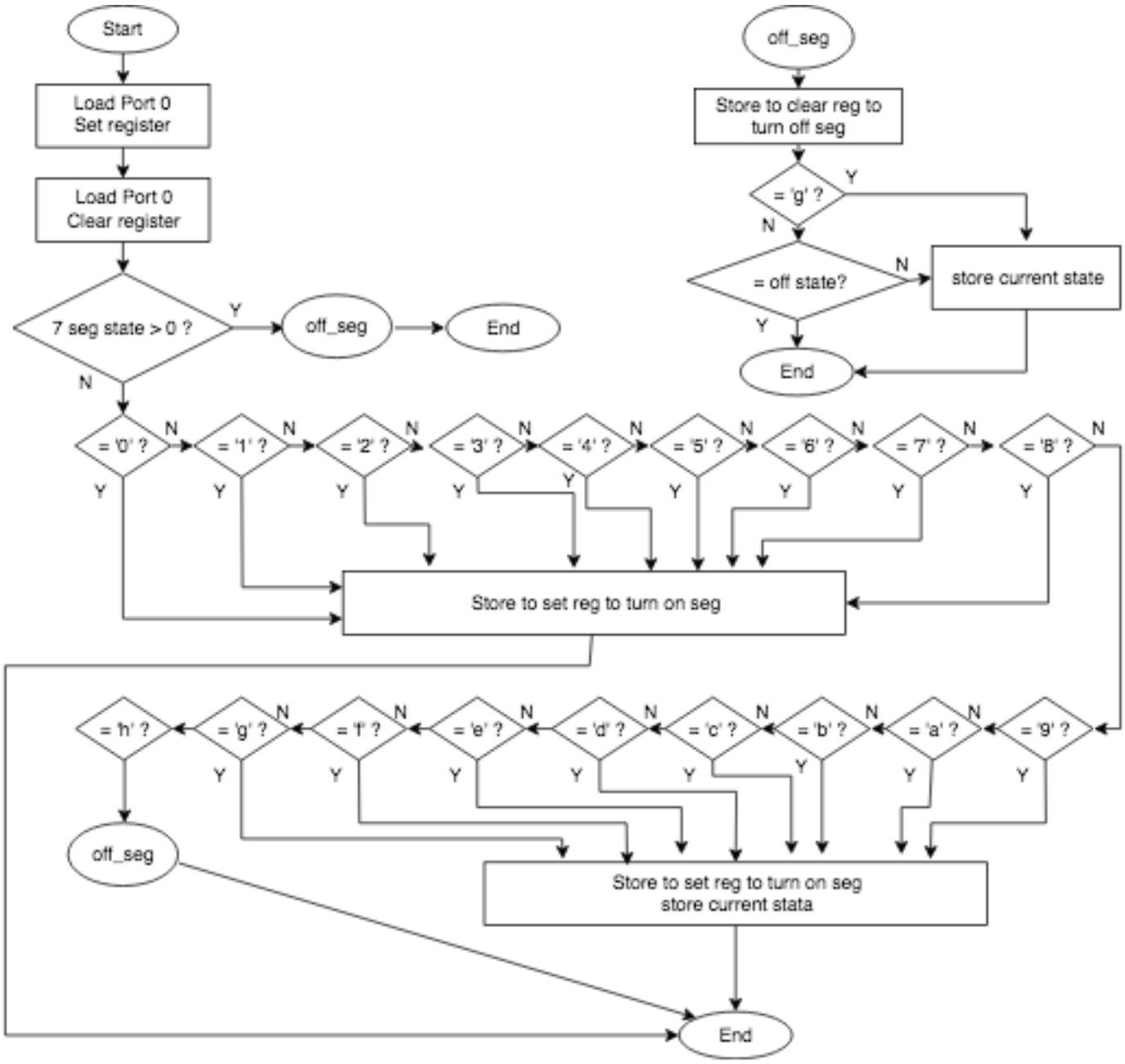
3.4 setup


Start
Load Port0 Direction register
Store to Direction register
Set Pin 5, 7, 8, 9, 10, 12, 14, 15
Initialize the input r0 as #71 (g segment)
Display digit on 7 segment
End

3.5 FIQ_Handler



3.6 display_digit_on_7_seg



4 Summary

There are three components to this program. The Cwrapper, library, and the main program itself. The first thing the program does is initializes pin select, UART0, and interrupts. The main program first sets up the initial state of the ARM board then prints the instructions on how to use the program. Afterwards the main program enters an infinite loop which waits for either a push button interrupt or a UART0 interrupt. After an interrupt happens it will branch to the FIQ interrupt handler. The handler first checks to see which type of interrupt occurred. The handler first checks EXTINT register's 2nd bit. If this bit is 0 it branches to the push button case of the handler. In the push button case it switches the sign of r7 which contains the on/off status of the display (negative for on and positive for off). It then loads the previous state held in r4 into r0 before branching to display digit on seven-segment display. In display digit on seven-segment display, it checks whether r7 is negative or positive, if it is positive it turns off the display. If r7 is negative it turns on and display the previous state before the display was turned off. After the display is either turned on or off the handler clears the interrupt before returning to the infinite loop. If the interrupt is not an external interrupt, the handler checks to see if it is a UART0 interrupt. It does this by checking U0IIR register's 0th bit. If the bit is 1 there is no pending interrupt exit handler, if the bit is 0 there is an interrupt pending branches to UART0 case for handler. The UART0 case first reads the character from the U0THR then echoes this character back to putty. The handler then checks to see if the character is q if it is end the program, if it isn't display character on the seven-segment display. After displaying the character, the handler returns to the infinite loop.

4.1 Interrupt_init:

Initializes the interrupts by changing bits of the respective registers. The interrupts enabled are UART0, and external interrupt 1.

4.2 FIQ_Handler:

Handles the interrupt by first differentiating between the interrupts before branching to branches associated with each interrupt. External interrupt turns the display on/off and UART0 interrupt displays hexadecimal entered via putty onto the display.

4.3 Display_digit_on_7_seg_display:

The display digit subroutine didn't change much from the previous lab with the only new addition being a new off state. The subroutine now also stores the previous state of the display, making sure never to store the off state. This makes it so that all putty inputs will be ignored when the display is off.

4.4 Setup:

The setup firstly, sets up the seven-segment by enabling it via editing values in IO0DIR. Set also initializes the state so that when the display is first turned on it only displays the g segment.