Embedded Systems Lab

CPE 325-02

Introduction to MSP430

By: David Thornton

Lab Date: August 20, 2020

Lab Due: August 31, 2020

Demonstration Due: August 31, 2020

Introduction

The purpose of this lab is to become familiar with function calls and the manipulation of ASCII values in the C programming language with debugging on the MSP-EXP430F5529LP using Code Composer Studio.

Theory

Topic 1: Function calls. Students should be able to inspect variables, set watchpoints, set and monitor breakpoints, monitor registers and memory, and show the output.

Topic 2: ASCII values. Students should be able to inspect variables, set watchpoints, set and monitor breakpoints, monitor registers and memory, and show the output.

Flow Chart:

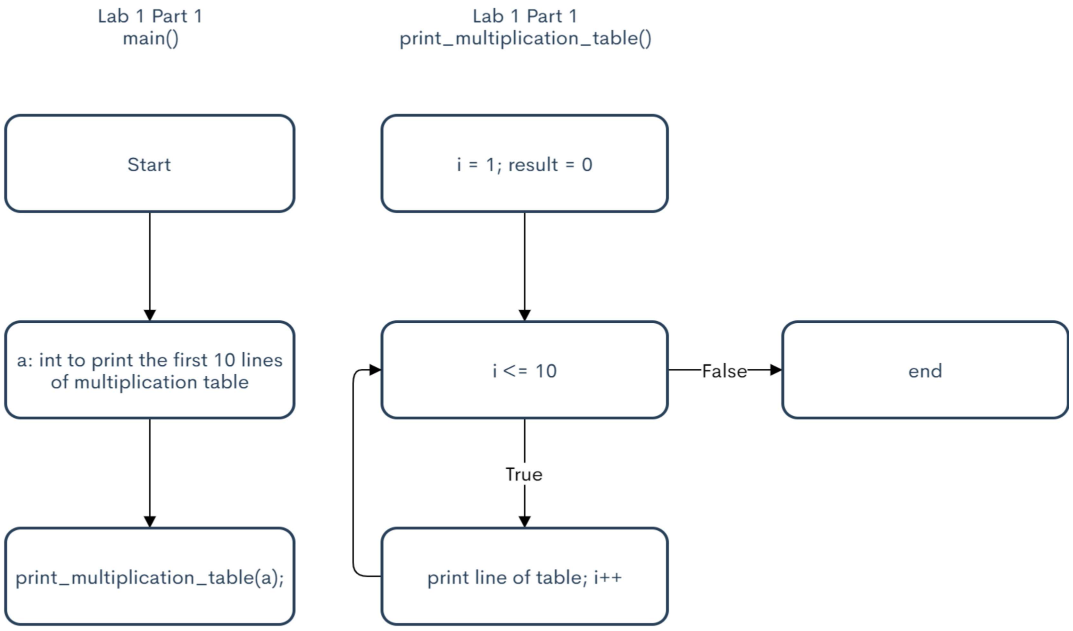
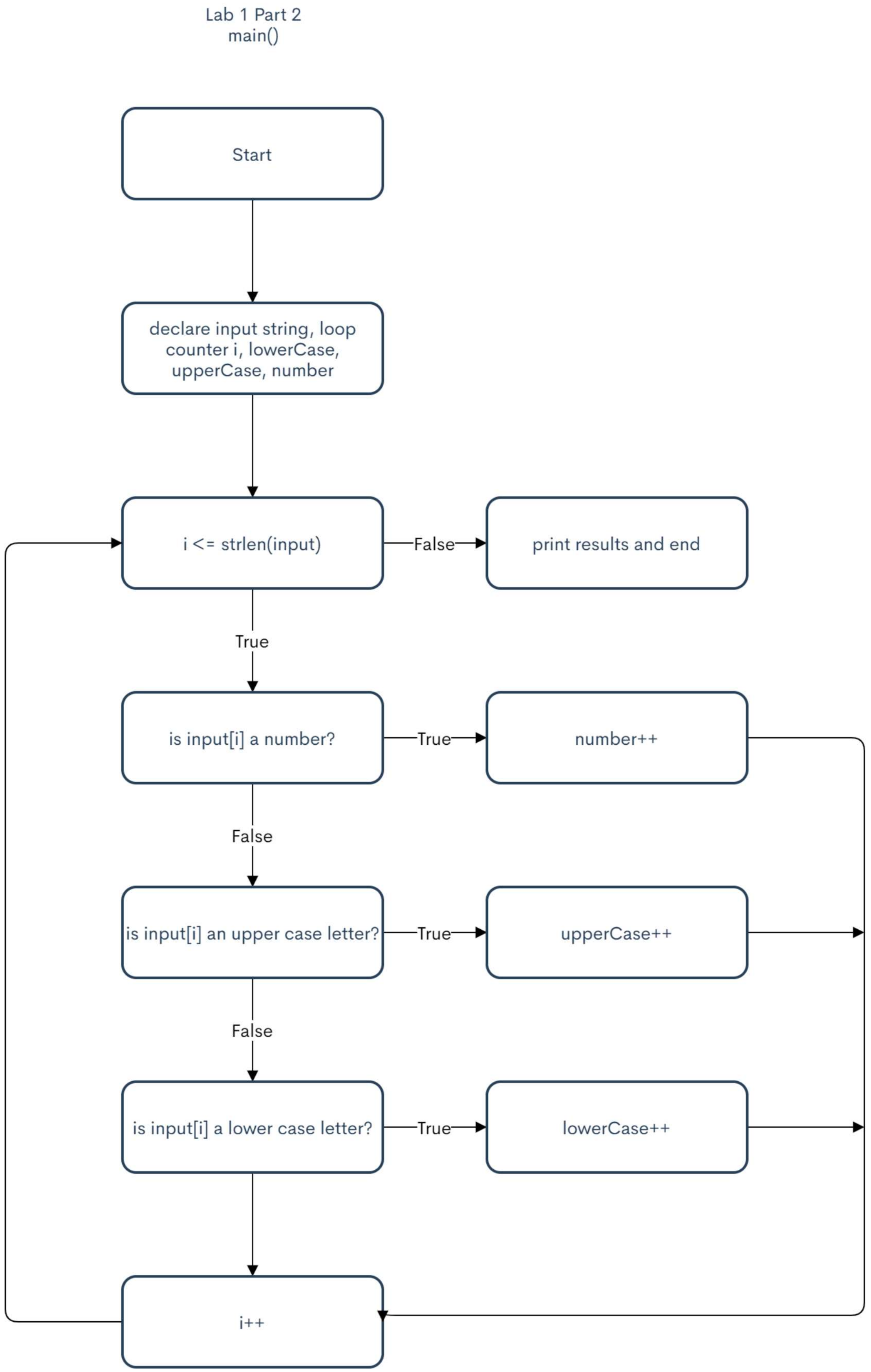

Lab 1 Part 1
main()
Lab 1 Part 1
print_multiplication_table()
Start
i = 1; result = 0
a: int to print the first 10 lines of multiplication table
i <= 10
False
end
True
print_multiplication_table(a);
print line of table; i++

Figure 1:

Lab 1 Part 1

Figure 2: Lab 1 Part 2



Observations

Upon stepping through the programs, I was able to see how the registers change with the “Registers” window and compare it to the code provided in the “Disassembly” window thus furthering my knowledge of assembly level code. For the small scope of this project, it was not necessary to watch any expressions.

Conclusion

This lab expanded my understanding of function calls and ASCII values, as well as increased my knowledge of the C programming language, Code Composer Studio, and how to run and debug a program on the MSP-EXP430F5529LP. The most significant issues I faced during this lab were not related to the lab material at all, rather they were related to being unfamiliar with Code Composer Studio. With the small scope of this lab, I was able to get a firm understanding of how to use the software for future labs. The mistakes I made in this lab will greatly reduce the debugging time of future labs. I am quite frustrated that the lab objectives are not clear, i.e. what will a student learn from this lab. This frustration was compounded even more so by the report template asking for the lab assignment question.

Demo link

Appendix

Appendix 1: Lab1_P1.c

```
/*--------------------------------------------------------
* File: Lab1_P1.c
* Description: Prints the first 10 lines of a
*            multiplication table for an unsigned integer.
* Input: Unsigned integer
* Output: Multiplication table
* Author: David Thornton
* Lab Section: 2
* Date: August 31, 2020
* *-----------------------------------------------------*/
#include <stdio.h> // For printf
#include <msp430.h>

void print_multiplication_table(unsigned int); // Function prototype

int main()
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer to prevent time out reset
            const unsigned int a = 25;
            print_multiplication_table(a);
            return 0;
}

void print_multiplication_table(unsigned int any_int)
{
    int i = 0; int result = 0;
    for (i = 1; i <= 10; i++)
    {
        result = any_int * i;
        printf("%d X %d = %d \n", any_int, i, result);
    }
}
```

Appendix 2: Lab1_P2.c

```
/*-------------------------------------------------------
* File: Lab1_P2.c
* Description: Counts the number of upper case, lower case,
*            and numbers (0-9) in a string.
* Input: String
* Output: Number of upper case, lower case, and numbers
* Author: David Thornton
* Lab Section: 2
* Date: August 31, 2020
* *-----------------------------------------------------*/
#include <stdio.h> // For printf
#include <string.h> // For strlen
#include <msp430.h>

int main()
{
   WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer to prevent time out reset
   const char input[] = "Hello! Welcome to CPE325. It’s a great day, isn’t it?";
   unsigned int i = 0; int lowerCase = 0; int upperCase = 0; int number = 0;
   for(i = 0; i<=strlen(input); i++)
   {
      if(input[i] >= 48 && input[i] <= 57) // [48 - 57] ASCII
         number++;
      else if(input[i] >= 65 && input[i] <= 90) // [65 - 90] ASCII
         upperCase++;
      else if(input[i] >= 97 && input[i] <= 122) // [97 - 122] ASCII
         lowerCase++;
   }
   printf("The string contains %d upper case and %d lower case characters and %d digits.",
upperCase, lowerCase, number);
   return 0;
}
```