C and PIC24 Assembly Language Programming

**5th Laboratory Report for ECE383**

**Microcomputers**

**Submitted by**

**Caleb Thompson**

**12145922**

**Orion Gregory**

**12160343**

**Group C**

**The University of Alabama**

**Tuscaloosa, Alabama 35487**

**Date**

**10/18/2023**

Abstract:

This lab provides a hands-on opportunity for students to apply their knowledge of C programming and PIC24 assembly language, familiarizing students with theoretical concepts with practical implementation of assembly and C.

Introduction:

The primary objective of this lab is to familiarize students with programming in C and its application in the context of PIC24 assembly language. The tasks within this lab involve the implementation of programming assignments using both the C language and the equivalent PIC24 assembly language.

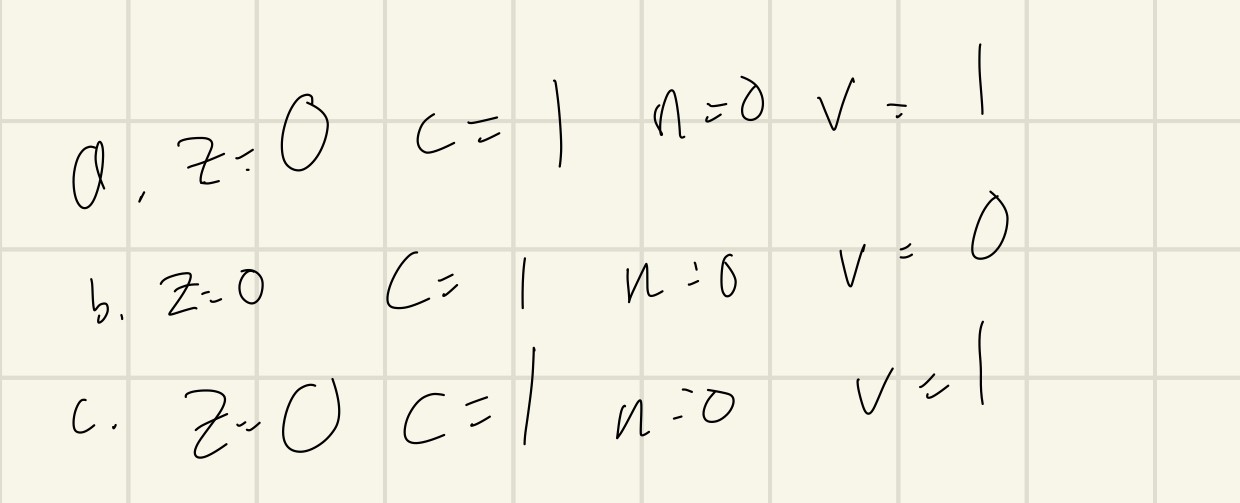
Procedure:

First, we created a new project within MPLAB and selected the PIC24HJ128GP502 device. We must include the microchip/lib/include directory, as this allows us to use it as a library. Then we must copy the given C program and name it as task1.c, we build the program and add the appropriate windows to our view (Program Memory, File Registers, Special Function Registers, and Watch). We then run the debugger and keep track of the status flags for each arithmetic operation, (Negative, Carry, Zero, Overflow).

Next, we created a new project named task2, and add a c file to the project called task2.c. We are tasked with creating a c program that counts the number of one bits in a 16-bit unsigned integer. We are given the value 0xF508 and told that the number of ones should be 7 and the first one occurs at the 3rd bit. We are to compare our C program’s output to this given correct output. Then for task 3, we are tasked with creating an assembly program that implements the same logic as task2,

Last, we are to convert a program written in PIC24 assembly and write it into C with the same logic and output.

Deliverables:

Deliverable 1:  


Deliverable 2:

Type casting was used in the line “u16\_d = (uint16) u8\_x + (uint16) u8\_y;”. This is done to ensure that the correct value is placed into u16\_d, as adding two 8-bit numbers can possibly lead to overflow, which would give an inaccurate result.

(fig 2) shows results of Task 1: MPLAB Introduction after changing avalue to 2047

Deliverable 3:  
A screenshot of a computer

Description automatically generated

Deliverable 4:  
A screenshot of a computer

Description automatically generated

Deliverable 6.

A screenshot of a computer

Description automatically generated

Deliverable 8.

A screenshot of a computer

Description automatically generated

Deliverable 10.

**This deliverable was done in person.**

Deliverable 5, 7, 9:  
These are code files and will be attached.

Conclusion:

In this lab we learned how to implement C code using MPLAB. We then learned how to convert assembly code into C code. Through the 4 tasks, students are asked to create programs in both C and assembly, and relate assembly code to C, and C to assembly.