MPLAB Introduction and PIC24 Assembly Language

**4th Laboratory Report for ECE383**

**Microcomputers**

**Submitted by**

**Caleb Thompson**

**12145922**

**Orion Gregory**

**12160343**

**Group C**

**The University of Alabama**

**Tuscaloosa, Alabama 35487**

**Date**

**10/12/2023**

Abstract:

The objective of this lab was to become familiar with the microchip MPLAB integrated design environment. This was done by writing and simulating simple PIC24 assembly language programs in MPLAB and then testing our results against hand calculations of the same operations to check for correctness.

Introduction:

This lab introduces students to the Microchip MPLAB Integrated Design Environment, focusing on simulation and implementation of PIC24 assembly language programs. All necessary files are in C:\microchip\chap3, and it is recommended to follow installation instructions on Blackboard for the required pic24\_code\_examples\_mplab.zip file. Using MPLAB allows us to simulate our microcontroller, and use a debugger to step through the program.

Procedure:

First, we copied the lab 4 files to a suitable directory. Then, we opened up the MPLAB IDE and used it to open an existing project from the lab 4 files called “mptst\_word.mcp”. Next, we selected the appropriate device from the configure menu and built the file. After opening all the appropriate watch windows, we stepped into the simulation and watched the variables change as we stepped through the program. We then changed the value of the avalue in the program to 2047 and resimulated following the same procedure.

Next, we saved the project as a new project called myadd, and using Orion’s CWID number, 12160343, we translated the given C code to assembly, built it, and simulated by stepping into and stepping through the program. We then hand calculated the values and compared the simulation results with our calculations to ensure correctness.

We then created a new project called mysub, and using Orion’s CWID number, 12160343, we translated the given C code to assembly, built it, and simulated by stepping into and through the program. We then hand calculated the values and compared the simulation results with our calculations to ensure correctness. Next, we hand calculated the expected flag values when j and l were added and subtracted and created a table to show the results of our calculations.

Last, we created a project called mylogicops, translated the logical operations in the given c code to assembly, and built and simulated our program. We then hand calculated the results and compared them with the results of our simulation to ensure correctness.

Deliverables:

Deliverable 1:   
A screenshot of a computer

Description automatically generated

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

Deliverable 2:

A screenshot of a computer

Description automatically generated

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

Deliverable 3, 6:

These are the .s files also sent.

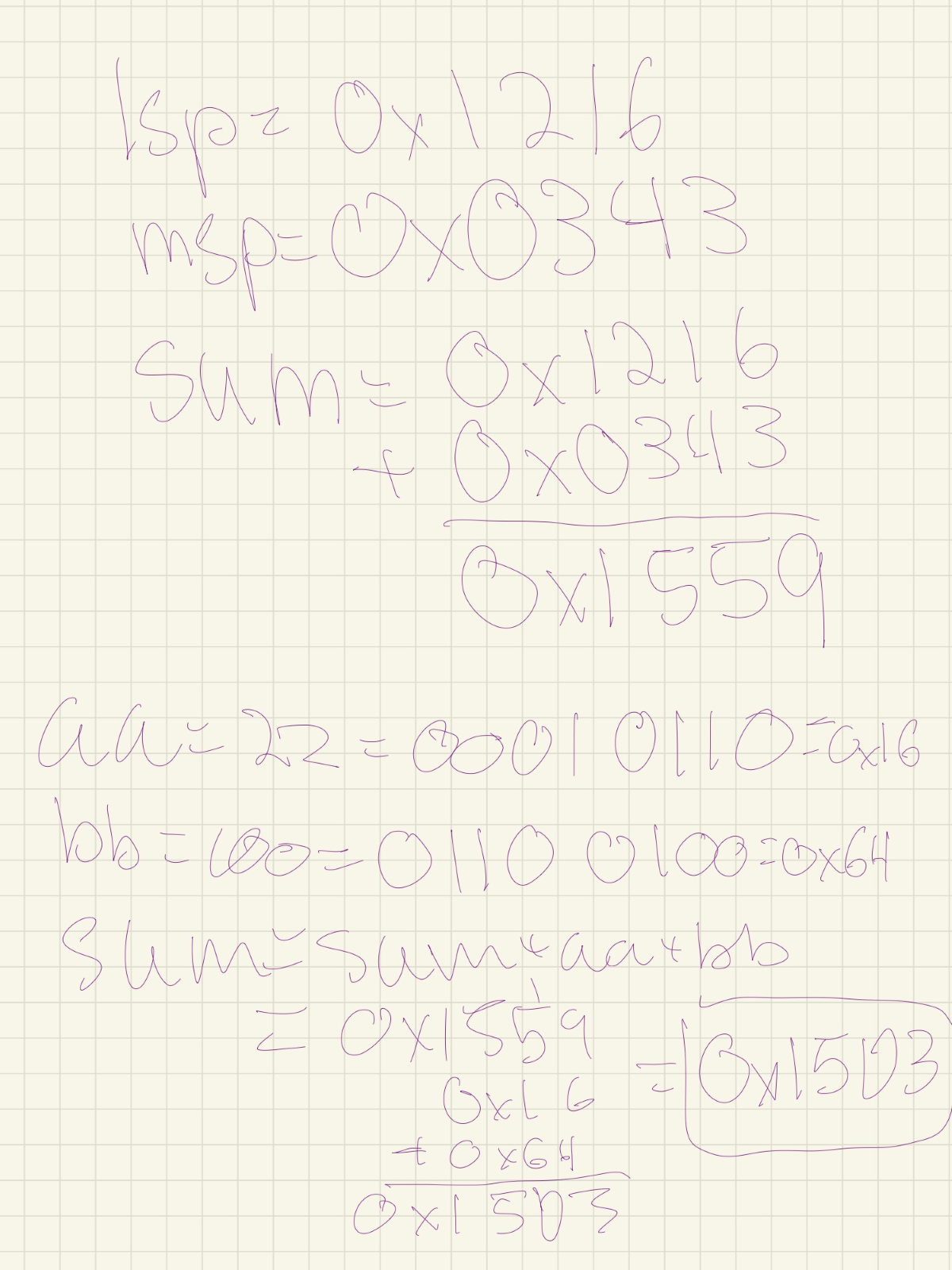
Deliverable 4:

A screenshot of a computer

Description automatically generated

(fig 4) shows results of Task 2: myadd using 12160343 as the CWID for the program

Deliverable 5.



Deliverable 7.   
A close-up of a math problem

Description automatically generated

Deliverable 8.

A screenshot of a computer

Description automatically generated

(fig 8) shows results of Task 3: mysub using 12150343 as the CWID for the program

Deliverable 9:

A math equations on a graph paper

Description automatically generated

Deliverable 10.

A math equations on a graph paper

Description automatically generated

(fig 10) shows hand calculations for Task 4: mylogicops

Deliverable 11.   
A screenshot of a computer

Description automatically generated

(fig 11) shows results of Task 4: mylogicops

Conclusion:

In this lab we learned how to open and create projects in MPLAB IDE. We learned how to write PIC24 assembly language in MPLAB and use watch windows to check our variables as we simulate programs using the IDE. This was done by opening and simulating existing projects, as well as editing existing projects, and creating our own projects by translating given C code into PIC24 assembly language.