MPLAB Introduction and PIC24 Assembly Language
4th Laboratory Report for ECE 383 Microcomputers

Submitted by
Patrick Brooks
Lauren Brackin
11650957

The University of Alabama
Tuscaloosa, AL 35487

2/19/21

Abstract:

The main objectives of this lab are to introduce students MPLAB and PCI24 assembly code. They will get a glimpse into the ecosystem. Students will carry these skills with them throughout their careers. Task one starts by getting students to familiarized with the interface of MPLAB. They will open up a file and modify it. Task two instructs students to create their own adder program called “myadd”. Task three is where students will create a subtraction program using different variables but generally the same approach. Task four tells students create the “mylogico” program that is written in C but must be converted to PIC24 assembly.

Introduction:

The objective of the lab is to introduce students to MPLAB and coding with PIC24 assembly. Task one focuses on getting students familiarized with the MPLAB software. In task two students are instructed create an adder program by converting the given C code into PIC24 assembly. Task three is where students will create a subtracting program by converting the given C code into PIC24 assembly. Task four instructs students to create a logic comparing program by converting the given C code into PIC24 assembly code.

Procedure/Results:

Task 1: MPLAB Introduction

a. Open "mptst_word.mcp" file from Chapt. 3 folder

i. Use Configure → Select Device → PIC24HJ128GP502

ii. Project → Build All

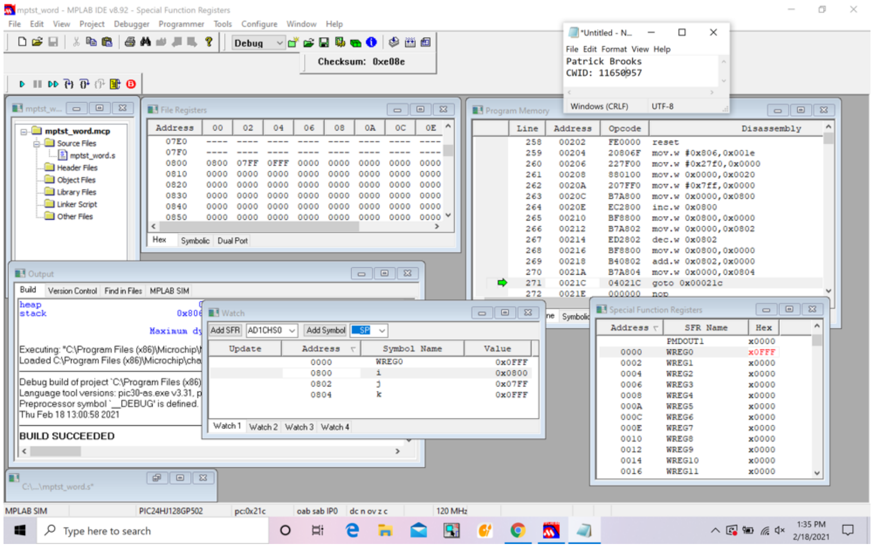
iii. View → Program Memory

iv. View → File Registers

v. View → Special Function Registers

vi. Debugger → Select Tool → MPLAB Sim

vii. Debugger → Step Over



```
mptst_word - MPLAB IDE v8.92 - Special Function Registers

Checksum:  0xe08e
```

```
*Untitled - N...
Patrick Brooks
CWID: 11650957
```

```
File Registers
Address   00    02    04    06    08    0A    0C    0E
  07E0   ----  ----  ----  ----  ----  ----  ----  ----
  07F0   ----  ----  ----  ----  ----  ----  ----  ----
  0800   0800  07FF  0FFF  0000  0000  0000  0000  0000
  0810   0000  0000  0000  0000  0000  0000  0000  0000
  0820   0000  0000  0000  0000  0000  0000  0000  0000
  0830   0000  0000  0000  0000  0000  0000  0000  0000
  0840   0000  0000  0000  0000  0000  0000  0000  0000
  0850   0000  0000  0000  0000  0000  0000  0000  0000
```

```
Program Memory
   Line  Address  Opcode  Disassembly
   258   00202    FE0000  reset
   259   00204    20806F  mov.w #0x806,0x001e
   260   00206    227F00  mov.w #0x27f0,0x0000
   261   00208    880100  mov.w 0x0000,0x0020
   262   0020A    207FF0  mov.w #0x7ff,0x0000
   263   0020C    B7A800  mov.w 0x0000,0x0800
   264   0020E    EC2800  inc.w 0x0800
   265   00210    BF8800  mov.w 0x0800,0x0000
   266   00212    B7A802  mov.w 0x0000,0x0802
   267   00214    ED2802  dec.w 0x0802
   268   00216    BF8800  mov.w 0x0800,0x0000
   269   00218    B40802  add.w 0x0802,0x0000
   270   0021A    B7A804  mov.w 0x0000,0x0804
 ⇨ 271   0021C    04021C  goto 0x00021c
   272   0021E    000000  nop
```

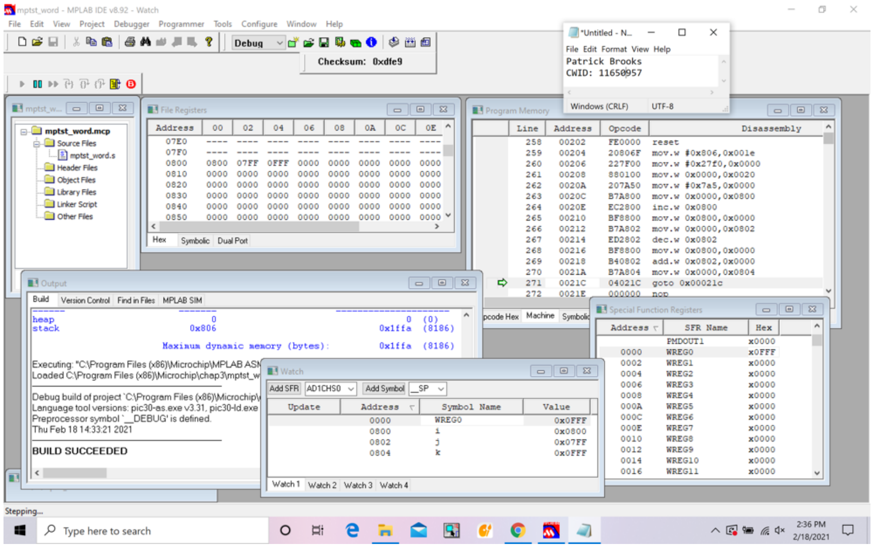
```
Output
heap                                  0
stack                             0x806
                     Maximum dy

Executing: "C:\Program Files (x86)\Microchip\
Loaded C:\Program Files (x86)\Microchip\cha

Debug build of project `C:\Program Files (x86)
Language tool versions: pic30-as.exe v3.31, p
Preprocessor symbol `__DEBUG' is defined.
Thu Feb 18 13:00:58 2021

BUILD SUCCEEDED
```

```
Watch
Update   Address   Symbol Name   Value
         0000      WREG0         0x0FFF
         0800      i             0x0800
         0802      j             0x07FF
         0804      k             0x0FFF
```

```
Special Function Registers
Address  SFR Name  Hex
         PMDOUT1   x0000
 0000    WREG0     x0FFF
 0002    WREG1     x0000
 0004    WREG2     x0000
 0006    WREG3     x0000
 0008    WREG4     x0000
 000A    WREG5     x0000
 000C    WREG6     x0000
 000E    WREG7     x0000
 0010    WREG8     x0000
 0012    WREG9     x0000
 0014    WREG10    x0000
 0016    WREG11    x0000
```

```
MPLAB SIM   PIC24HJ128GP502   pc:0x21c   oab sab IP0   dc n ov z c   120 MHz
```

Figure 1. Deliverable 1



```
mptst_word - MPLAB IDE v8.92 - Watch

Checksum:  0xdfe9
```

```
*Untitled - N...
Patrick Brooks
CWID: 11650957
```

```
File Registers
Address   00    02    04    06    08    0A    0C    0E
  07E0   ----  ----  ----  ----  ----  ----  ----  ----
  07F0   ----  ----  ----  ----  ----  ----  ----  ----
  0800   0800  07FF  0FFF  0000  0000  0000  0000  0000
  0810   0000  0000  0000  0000  0000  0000  0000  0000
  0820   0000  0000  0000  0000  0000  0000  0000  0000
  0830   0000  0000  0000  0000  0000  0000  0000  0000
  0840   0000  0000  0000  0000  0000  0000  0000  0000
  0850   0000  0000  0000  0000  0000  0000  0000  0000
```

```
Program Memory
   Line  Address  Opcode  Disassembly
   258   00202    FE0000  reset
   259   00204    20806F  mov.w #0x806,0x001e
   260   00206    227F00  mov.w #0x27f0,0x0000
   261   00208    880100  mov.w 0x0000,0x0020
   262   0020A    207A50  mov.w #0x7a5,0x0000
   263   0020C    B7A800  mov.w 0x0000,0x0800
   264   0020E    EC2800  inc.w 0x0800
   265   00210    BF8800  mov.w 0x0800,0x0000
   266   00212    B7A802  mov.w 0x0000,0x0802
   267   00214    ED2802  dec.w 0x0802
   268   00216    BF8800  mov.w 0x0800,0x0000
   269   00218    B40802  add.w 0x0802,0x0000
   270   0021A    B7A804  mov.w 0x0000,0x0804
 ⇨ 271   0021C    04021C  goto 0x00021c
   272   0021E    000000  nop
```

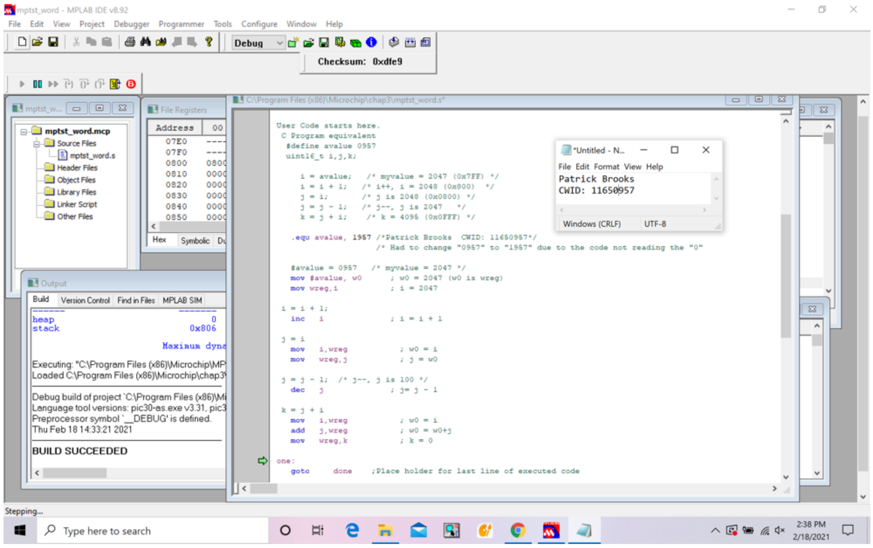
```
Output
-------               -------                    ---------------------
heap                        0                                  0  (0)
stack                   0x806                             0x1ffa  (8186)

                     Maximum dynamic memory (bytes):      0x1ffa  (8186)

Executing: "C:\Program Files (x86)\Microchip\MPLAB ASM
Loaded C:\Program Files (x86)\Microchip\chap3\mptst_w
--------------------------------------------------------
Debug build of project `C:\Program Files (x86)\Microchip\
Language tool versions: pic30-as.exe v3.31, pic30-ld.exe
Preprocessor symbol `__DEBUG' is defined.
Thu Feb 18 14:33:21 2021
--------------------------------------------------------
BUILD SUCCEEDED
```

```
Watch
Update   Address   Symbol Name   Value
         0000      WREG0         0x0FFF
         0800      i             0x0800
         0802      j             0x07FF
         0804      k             0x0FFF
```

```
Special Function Registers
Address  SFR Name  Hex
         PMDOUT1   x0000
 0000    WREG0     x0FFF
 0002    WREG1     x0000
 0004    WREG2     x0000
 0006    WREG3     x0000
 0008    WREG4     x0000
 000A    WREG5     x0000
 000C    WREG6     x0000
 000E    WREG7     x0000
 0010    WREG8     x0000
 0012    WREG9     x0000
 0014    WREG10    x0000
 0016    WREG11    x0000
```

```
Stepping...
```

Figure 2. Deliverable 2



mptst_word - MPLAB IDE v8.92

Checksum: 0xdfe9

C:\Program Files (x86)\Microchip\chap3\mptst_word.s*

```
User Code starts here.
 C Program equivalent
  #define avalue 0957
  uint16_t i,j,k;

     i = avalue;   /* myvalue = 2047 (0x7FF) */
     i = i + 1;   /* i++, i = 2048 (0x800)  */
     j = i;       /* j is 2048 (0x0800) */
     j = j - 1;   /* j--, j is 2047   */
     k = j + i;    /* k = 4095 (0x0FFF) */

   .equ avalue, 1957 /*Patrick Brooks  CWID: 11650957*/
                    /* Had to change "0957" to "1957" due to the code not reading the "0"

   #avalue = 0957   /* myvalue = 2047 */
   mov #avalue, w0       ; w0 = 2047 (w0 is wreg)
   mov wreg,i            ; i = 2047

 i = i + 1;
   inc   i               ; i = i + 1

 j = i
   mov   i,wreg            ; w0 = i
   mov   wreg,j            ; j = w0

 j = j - 1;  /* j--, j is 100 */
   dec   j               ; j= j - 1

 k = j + i
   mov   i,wreg            ; w0 = i
   add   j,wreg            ; w0 = w0+j
   mov   wreg,k            ; k = 0

one:
   goto     done    ;Place holder for last line of executed code
```

*Untitled - N...

```
Patrick Brooks
CWID: 11650957
```

mptst_word.mcp: Source Files (mptst_word.s), Header Files, Object Files, Library Files, Linker Script, Other Files

File Registers

```
Address   00
 07E0    ----
 07F0    ----
 0800    0800
 0810    0000
 0820    0000
 0830    0000
 0840    0000
 0850    0000
```

Output

```
heap                          0
stack                     0x806

                    Maximum dyna

Executing: "C:\Program Files (x86)\Microchip\MP
Loaded C:\Program Files (x86)\Microchip\chap3\

Debug build of project `C:\Program Files (x86)\Mi
Language tool versions: pic30-as.exe v3.31, pic3
Preprocessor symbol `__DEBUG' is defined.
Thu Feb 18 14:33:21 2021

BUILD SUCCEEDED
```

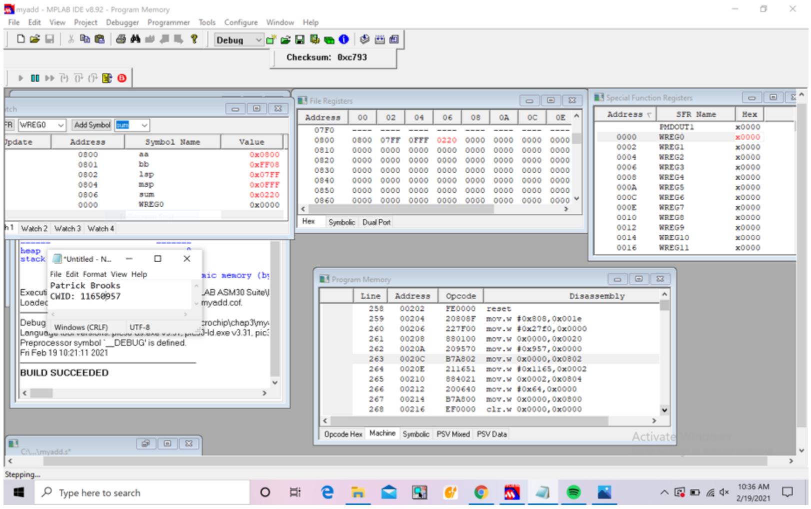
Stepping...

Figure 3. Deliverable 3

Task 2: Addition Program

a. Create "myadd" file

i. Refer to lab instructions for C code to translate to PIC24 assembly code



myadd - MPLAB IDE v8.92 - Program Memory

Checksum: 0xc793

Watch

```
Update   Address   Symbol Name   Value
          0800      aa           0x0800
          0801      bb           0xFF08
          0802      lsp          0x07FF
          0804      msp          0x0FFF
          0806      sum          0x0220
          0000      WREG0        0x0000
```

File Registers

```
Address   00    02    04    06    08    0A    0C    0E
 07F0    ----  ----  ----  ----  ----  ----  ----  ----
 0800    0800  07FF  0FFF  0220  0000  0000  0000  0000
 0810    0000  0000  0000  0000  0000  0000  0000  0000
 0820    0000  0000  0000  0000  0000  0000  0000  0000
 0830    0000  0000  0000  0000  0000  0000  0000  0000
 0840    0000  0000  0000  0000  0000  0000  0000  0000
 0850    0000  0000  0000  0000  0000  0000  0000  0000
 0860    0000  0000  0000  0000  0000  0000  0000  0000
```

Special Function Registers

```
Address   SFR Name   Hex
          PMDOUT1    x0000
 0000     WREG0      x0000
 0002     WREG1      x0000
 0004     WREG2      x0000
 0006     WREG3      x0000
 0008     WREG4      x0000
 000A     WREG5      x0000
 000C     WREG6      x0000
 000E     WREG7      x0000
 0010     WREG8      x0000
 0012     WREG9      x0000
 0014     WREG10     x0000
 0016     WREG11     x0000
```

Program Memory

```
Line   Address   Opcode   Disassembly
 258   00202     FE0000   reset
 259   00204     20808F   mov.w #0x808,0x001e
 260   00206     227F00   mov.w #0x27f0,0x0000
 261   00208     880100   mov.w 0x0000,0x0020
 262   0020A     209570   mov.w #0x957,0x0000
 263   0020C     B7A802   mov.w 0x0000,0x0802
 264   0020E     211651   mov.w #0x1165,0x0002
 265   00210     884021   mov.w 0x0002,0x0804
 266   00212     200640   mov.w #0x64,0x0000
 267   00214     B7A800   mov.w 0x0000,0x0800
 268   00216     EF0000   clr.w 0x0000,0x0000
```

*Untitled - N...

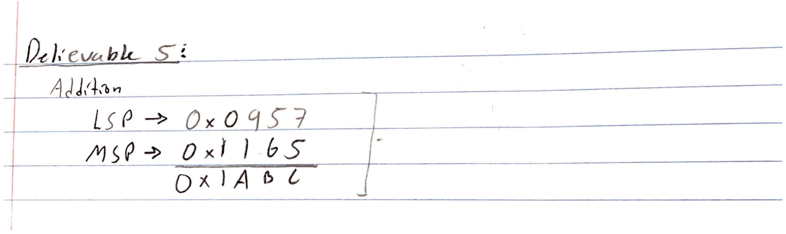
```
Patrick Brooks
CWID: 11650957
```

```
heap
stack
                                      mic memory (by
Executi                               LAB ASM30 Suite\
Loadec                                myadd.cof.

Debug                                 crochip\chap3\my
Language                              0-ld.exe v3.31, pic3
Preprocessor symbol `__DEBUG' is defined.
Fri Feb 19 10:21:11 2021

BUILD SUCCEEDED
```

C:\...\myadd.s*

Stepping...

Figure 4. Deliverable 4



Delievable 5:

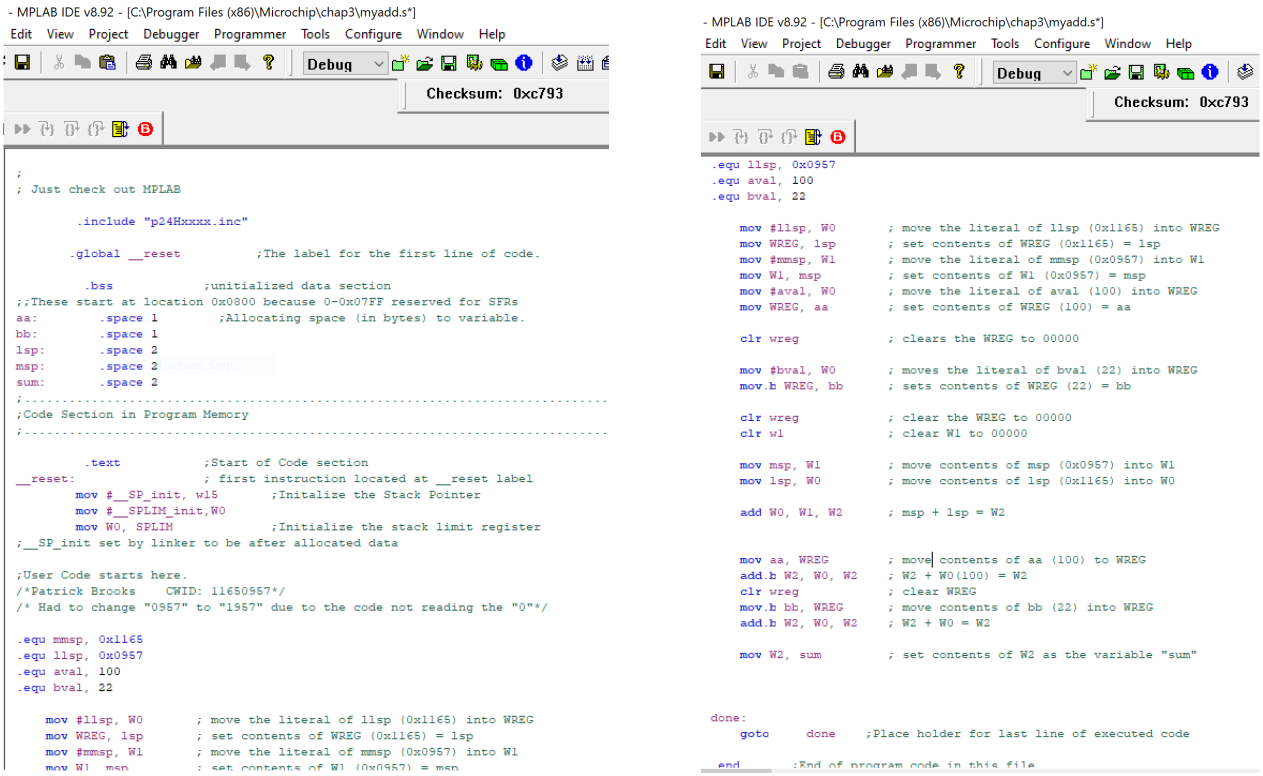
Addition

LSP → 0x0957

MSP → 0x1165

0x1ABC

Figure 5. Deliverable 5



- MPLAB IDE v8.92 - [C:\Program Files (x86)\Microchip\chap3\myadd.s*]
Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xc793

```
;
; Just check out MPLAB

        .include "p24Hxxxx.inc"

       .global __reset          ;The label for the first line of code.

         .bss             ;unitialized data section
;;These start at location 0x0800 because 0-0x07FF reserved for SFRs
aa:         .space 1        ;Allocating space (in bytes) to variable.
bb:         .space 1
lsp:        .space 2
msp:        .space 2
sum:        .space 2
;..............................................................................
;Code Section in Program Memory
;..............................................................................

         .text            ;Start of Code section
__reset:                  ; first instruction located at __reset label
        mov #__SP_init, w15       ;Initalize the Stack Pointer
        mov #__SPLIM_init,W0
        mov W0, SPLIM             ;Initialize the stack limit register
;__SP_init set by linker to be after allocated data

;User Code starts here.
/*Patrick Brooks    CWID: 11650957*/
/* Had to change "0957" to "1957" due to the code not reading the "0"*/

.equ mmsp, 0x1165
.equ llsp, 0x0957
.equ aval, 100
.equ bval, 22

    mov #llsp, W0       ; move the literal of llsp (0x1165) into WREG
    mov WREG, lsp       ; set contents of WREG (0x1165) = lsp
    mov #mmsp, W1       ; move the literal of mmsp (0x0957) into W1
    mov W1, msp         ; set contents of W1 (0x0957) = msp
```

- MPLAB IDE v8.92 - [C:\Program Files (x86)\Microchip\chap3\myadd.s*]
Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xc793

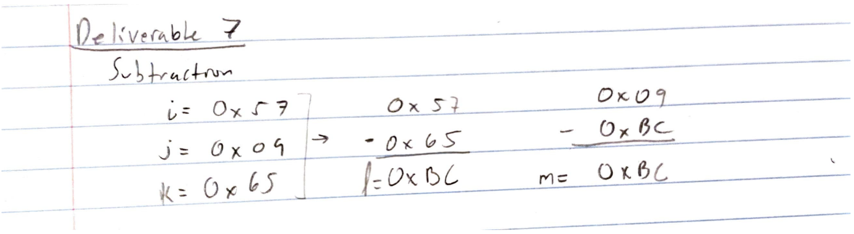
```
.equ llsp, 0x0957
.equ aval, 100
.equ bval, 22

    mov #llsp, W0       ; move the literal of llsp (0x1165) into WREG
    mov WREG, lsp       ; set contents of WREG (0x1165) = lsp
    mov #mmsp, W1       ; move the literal of mmsp (0x0957) into W1
    mov W1, msp         ; set contents of W1 (0x0957) = msp
    mov #aval, W0       ; move the literal of aval (100) into WREG
    mov WREG, aa        ; set contents of WREG (100) = aa

    clr wreg            ; clears the WREG to 00000

    mov #bval, W0       ; moves the literal of bval (22) into WREG
    mov.b WREG, bb      ; sets contents of WREG (22) = bb

    clr wreg            ; clear the WREG to 00000
    clr wl              ; clear W1 to 00000

    mov msp, W1         ; move contents of msp (0x0957) into W1
    mov lsp, W0         ; move contents of lsp (0x1165) into W0

    add W0, W1, W2      ; msp + lsp = W2


    mov aa, WREG        ; move contents of aa (100) to WREG
    add.b W2, W0, W2    ; W2 + W0(100) = W2
    clr wreg            ; clear WREG
    mov.b bb, WREG      ; move contents of bb (22) into WREG
    add.b W2, W0, W2    ; W2 + W0 = W2

    mov W2, sum         ; set contents of W2 as the variable "sum"


done:
    goto     done    ;Place holder for last line of executed code

 end       ;End of program code in this file
```

Figure 6. Deliverable 6

<u>Task 3</u>: Subtraction Program

a. Create “mysub” file

i. Refer to lab instructions for C code to translate to PIC24 assembly code



Deliverable 7

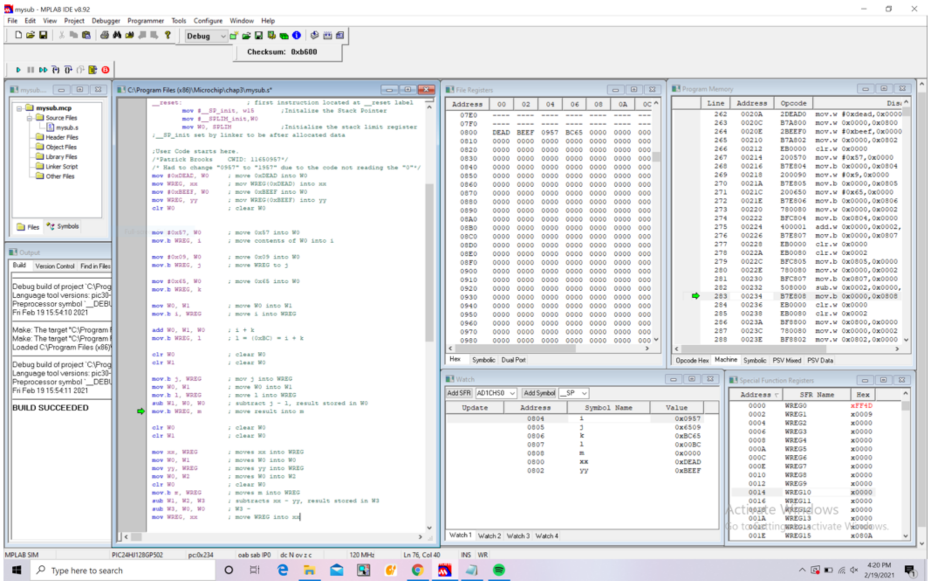
Subtraction

i = 0x57   0x57   0x09

j = 0x09 → - 0x65   - 0xBC

k = 0x65   l = 0xBC   m = 0xBC

Figure 7. Deliverable 7



mysub - MPLAB IDE v8.92

Checksum: 0xb600

C:\Program Files (x86)\Microchip\chap3\mysub.s*

```
__reset:                    ; first instruction located at __reset label
        mov #__SP_init, w15       ;Initalize the Stack Pointer
        mov #__SPLIM_init,W0
        mov W0, SPLIM             ;Initialize the stack limit register
;__SP_init set by linker to be after allocated data

;User Code starts here.
/*Patrick Brooks    CWID: 11650957*/
/* Had to change "0957" to "1957" due to the code not reading the "0"*/
mov #0xDEAD, W0     ; move 0xDEAD into W0
mov WREG, xx        ; mov WREG(0xDEAD) into xx
mov #0xBEEF, W0     ; move 0xBEEF into W0
mov WREG, yy        ; mov WREG(0xBEEF) into yy
clr W0              ; clear W0


mov #0x57, W0       ; move 0x57 into W0
mov.b WREG, i       ; move contents of W0 into i

mov #0x09, W0       ; move 0x09 into W0
mov.b WREG, j       ; move WREG to j

mov #0x65, W0       ; move 0x65 into W0
mov.b WREG, k

mov W0, W1          ; move W0 into W1
mov.b i, WREG       ; move i into WREG

add W0, W1, W0      ; i + k
mov.b WREG, l       ; l = (0xBC) = i + k

clr W0              ; clear W0
clr W1              ; clear W0

mov.b j, WREG       ; mov j into WREG
mov W0, W1          ; move W0 into W1
mov.b l, WREG       ; move l into WREG
sub W1, W0, W0      ; subtract j - l, result stored in W0
mov.b WREG, m       ; move result into m

clr W0              ; clear W0
clr W1              ; clear W0

mov xx, WREG        ; moves xx into WREG
mov W0, W1          ; moves W0 into W0
mov yy, WREG        ; moves yy into WREG
mov W0, W2          ; moves W0 into W2
clr W0              ; clear W0
mov.b m, WREG       ; moves m into WREG
sub W1, W2, W3      ; subtracts xx - yy, result stored in W3
sub W3, W0, W0      ; W3 -
mov WREG, xx        ; move WREG into xx
```

Output

```
Debug build of project `C:\Prog
Language tool versions: pic30-
Preprocessor symbol `__DEBU
Fri Feb 19 15:54:10 2021

Make: The target "C:\Program
Make: The target "C:\Program
Loaded C:\Program Files (x86)\

Debug build of project `C:\Prog
Language tool versions: pic30-
Preprocessor symbol `__DEBU
Fri Feb 19 15:54:11 2021

BUILD SUCCEEDED
```

File Registers

```
Address  00    02    04    06    08    0A    0C
 07E0    ----  ----  ----  ----  ----  ----  ---
 07F0    ----  ----  ----  ----  ----  ----  ---
 0800    DEAD  BEEF  0957  BC65  0000  0000  000
 0810    0000  0000  0000  0000  0000  0000  000
 0820    0000  0000  0000  0000  0000  0000  000
 0830    0000  0000  0000  0000  0000  0000  000
 0840    0000  0000  0000  0000  0000  0000  000
 0850    0000  0000  0000  0000  0000  0000  000
 0860    0000  0000  0000  0000  0000  0000  000
 0870    0000  0000  0000  0000  0000  0000  000
 0880    0000  0000  0000  0000  0000  0000  000
 0890    0000  0000  0000  0000  0000  0000  000
 08A0    0000  0000  0000  0000  0000  0000  000
 08B0    0000  0000  0000  0000  0000  0000  000
 08C0    0000  0000  0000  0000  0000  0000  000
 08D0    0000  0000  0000  0000  0000  0000  000
 08E0    0000  0000  0000  0000  0000  0000  000
 08F0    0000  0000  0000  0000  0000  0000  000
 0900    0000  0000  0000  0000  0000  0000  000
 0910    0000  0000  0000  0000  0000  0000  000
 0920    0000  0000  0000  0000  0000  0000  000
 0930    0000  0000  0000  0000  0000  0000  000
 0940    0000  0000  0000  0000  0000  0000  000
 0950    0000  0000  0000  0000  0000  0000  000
 0960    0000  0000  0000  0000  0000  0000  000
 0970    0000  0000  0000  0000  0000  0000  000
 0980    0000  0000  0000  0000  0000  0000  000
```

Program Memory

```
 Line  Address  Opcode   Dis
 262   0020A    2DEAD0   mov.w #0xdead,0x0000
 263   0020C    B7A800   mov.w 0x0000,0x0800
 264   0020E    2BEEF0   mov.w #0xbeef,0x0000
 265   00210    B7A802   mov.w 0x0000,0x0802
 266   00212    EB0000   clr.w 0x0000
 267   00214    200570   mov.w #0x57,0x0000
 268   00216    B7E804   mov.b 0x0000,0x0804
 269   00218    200090   mov.w #0x9,0x0000
 270   0021A    B7E805   mov.b 0x0000,0x0805
 271   0021C    200650   mov.w #0x65,0x0000
 272   0021E    B7E806   mov.b 0x0000,0x0806
 273   00220    780080   mov.w 0x0000,0x0002
 274   00222    BFC804   mov.b 0x0804,0x0000
 275   00224    400001   add.w 0x0000,0x0002,
 276   00226    B7E807   mov.b 0x0000,0x0807
 277   00228    EB0000   clr.w 0x0000
 278   0022A    EB0080   clr.w 0x0002
 279   0022C    BFC805   mov.b 0x0805,0x0000
 280   0022E    780080   mov.w 0x0000,0x0002
 281   00230    BFC807   mov.b 0x0807,0x0000
 282   00232    508000   sub.w 0x0002,0x0000,
 283   00234    B7E808   mov.b 0x0000,0x0808
 284   00236    EB0000   clr.w 0x0000
 285   00238    EB0080   clr.w 0x0002
 286   0023A    BF8800   mov.w 0x0800,0x0000
 287   0023C    780080   mov.w 0x0000,0x0002
 288   0023E    BF8802   mov.w 0x0802,0x0000
```

Watch

Add SFR: AD1CH50   Add Symbol: __SP

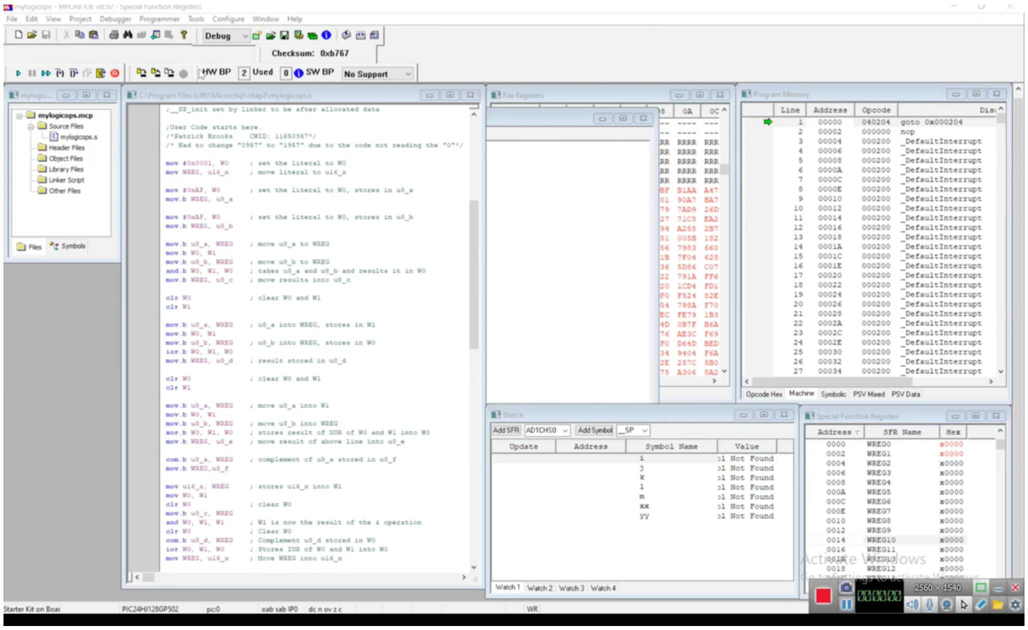
Update	Address	Symbol Name	Value
	0804	i	0x0957
	0805	j	0x6509
	0806	k	0xBC65
	0807	l	0x00BC
	0808	m	0x0000
	0800	xx	0xDEAD
	0802	yy	0xBEEF

Special Function Registers

Address	SFR Name	Hex
0000	WREG0	xFF4D
0002	WREG1	x0009
0004	WREG2	x0000
0006	WREG3	x0000
0008	WREG4	x0000
000A	WREG5	x0000
000C	WREG6	x0000
000E	WREG7	x0000
0010	WREG8	x0000
0012	WREG9	x0000
0014	WREG10	x0000
0016	WREG11	x0000
0018	WREG12	x0000
001A	WREG13	x0000
001C	WREG14	x0000
001E	WREG15	x080A

MPLAB SIM   PIC24HJ128GP502   pc:0x234   oab sab IP0   dc N ov z c   120 MHz   Ln 76, Col 40   INS   WR

Figure 8. Deliverable 8



```
;__SP_init set by linker to be after allocated data

;User Code starts here.
/*Patrick Brooks    CWID: 11650957*/
/* Had to change "0957" to "1957" due to the code not reading the "0"*/

mov #0x0001, W0       ; set the literal to W0
mov WREG, u16_x       ; move literal to u16_x

mov #0xAF, W0         ; set the literal to W0, stores in u8_a
mov.b WREG, u8_a

mov #0xAF, W0         ; set the literal to W0, stores in u8_b
mov.b WREG, u8_b

mov.b u8_a, WREG      ; move u8_a to WREG
mov.b W0, W1
mov.b u8_b, WREG      ; move u8_b to WREG
and.b W0, W1, W0      ; takes u8_a and u8_b and results it in W0
mov.b WREG, u8_c      ; move results into u8_c

clr W0                ; clear W0 and W1
clr W1

mov.b u8_a, WREG      ; u8_a into WREG, stores in W1
mov.b W0, W1
mov.b u8_b, WREG      ; u8_b into WREG, stores in W0
ior.b W0, W1, W0
mov.b WREG, u8_d      ; result stored in u8_d

clr W0                ; clear W0 and W1
clr W1

mov.b u8_a, WREG      ; move u8_a into W1
mov.b W0, W1
mov.b u8_b, WREG      ; move u8_b into WREG
xor.b W0, W1, W0      ; stores result of XOR of W0 and W1 into W0
mov.b WREG, u8_e      ; move result of above line into u8_e

com.b u8_a, WREG      ; complement of u8_a stored in u8_f
mov.b WREG,u8_f

mov u16_x, WREG       ; stores u16_x into W1
mov W0, W1
clr W0                ; clear W0
mov.b u8_c, WREG
and W0, W1, W1        ; W1 is now the result of the & operation
clr W0                ; Clear W0
com.b u8_d, WREG      ; Complement u8_d stored in W0
ior W0, W1, W0        ; Stores IOR of W0 and W1 into W0
mov WREG, u16_x       ; Move WREG into u16_x
```

Checksum: 0xb767

Line	Address	Opcode	Disassembly
1	00000	040204	goto 0x000204
2	00002	000000	nop
3	00004	000200	_DefaultInterrupt
4	00006	000200	_DefaultInterrupt
5	00008	000200	_DefaultInterrupt
6	0000A	000200	_DefaultInterrupt
7	0000C	000200	_DefaultInterrupt
8	0000E	000200	_DefaultInterrupt
9	00010	000200	_DefaultInterrupt
10	00012	000200	_DefaultInterrupt
11	00014	000200	_DefaultInterrupt
12	00016	000200	_DefaultInterrupt
13	00018	000200	_DefaultInterrupt
14	0001A	000200	_DefaultInterrupt
15	0001C	000200	_DefaultInterrupt
16	0001E	000200	_DefaultInterrupt
17	00020	000200	_DefaultInterrupt
18	00022	000200	_DefaultInterrupt
19	00024	000200	_DefaultInterrupt
20	00026	000200	_DefaultInterrupt
21	00028	000200	_DefaultInterrupt
22	0002A	000200	_DefaultInterrupt
23	0002C	000200	_DefaultInterrupt
24	0002E	000200	_DefaultInterrupt
25	00030	000200	_DefaultInterrupt
26	00032	000200	_DefaultInterrupt
27	00034	000200	_DefaultInterrupt

Add SFR AD1CHS0 Add Symbol __SP

Update	Address	Symbol Name	Value
		i	ol Not Found
		j	ol Not Found
		k	ol Not Found
		l	ol Not Found
		m	ol Not Found
		xx	ol Not Found
		yy	ol Not Found

Address	SFR Name	Hex
0000	WREG0	x0000
0002	WREG1	x0000
0004	WREG2	x0000
0006	WREG3	x0000
0008	WREG4	x0000
000A	WREG5	x0000
000C	WREG6	x0000
000E	WREG7	x0000
0010	WREG8	x0000
0012	WREG9	x0000
0014	WREG10	x0000
0016	WREG11	x0000
001A	WREG13	x0000
0018	WREG12	x0000

PIC24HJ128GP502 pc:0 oab sab IP0 dc n ov z c

mylogicops - MPLAB IDE v8.92 - Special Function Registers
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
HW BP 2 Used 0 SW BP No Support
mylogicops.mcp
Source Files
mylogicops.s
Header Files
Object Files
Library Files
Linker Script
Other Files
Files Symbols
C:\Program Files (x86)\Microchip\chap3\mylogicops.s
File Registers
Program Memory
Opcode Hex Machine Symbolic PSV Mixed PSV Data
Watch
Watch 1 Watch 2 Watch 3 Watch 4
Special Function Registers
Starter Kit on Boar
WR
Activate Windows
2560 × 1540

Figure 11. Deliverable 11

https://www.youtube.com/watch?v=eM2qSFXHAmE&feature=youtu.be&ab_channel=PatrickBrooks

Figure 12. Deliverable 12

Conclusion:

After completion of the lab students learned how to use MPLAB as well as how to code PIC24 assembly. As well as revisiting useful addition and subtraction concepts using binary/hex. Students also deepened their understanding of the lab report format, which will be used throughout their academic endeavors.