EE 2004
2020-2021: Semester B
Assignment 1
Due: Feb. 19, 2021

Instructions:

The .asm and .lst files for the programs required for Questions 1 and 2 should be submitted. The answers for Questions 3 and 4 should be provided in a Microsoft Word file. **The Microsoft Word file and all the requested .asm/.lst files should be zipped as a single zip file for submission.**

Students must submit the assignment through Canvas. Click on the item “Assignment” on the left panel. You should see a row with title “Assignment 1”. Click on the “Assignment 1” label and find the “Submit Assignment” label on the right panel. Click on it and upload the requested zip file.

Question 1 (30 marks)

Loop: Summing numbers in a sequence

A sequence is defined by the following recurrence relation:

$Q_n = Q_{n-1} + Q_{n-2} + Q_{n-3}$

with seed value

$Q_0 = 0$, $Q_1 = 0$ and $Q_2 = 1$ where Q_n is the nth number in the sequence.

Write a loop in assembly language that calculates the sum of the series $Q_0 + Q_1 + \ldots\ldots Q_{10}$. Store the sum in the file register with address 0x000. Save your program in a file named `Loop.asm`. Assemble and verify your code using MPLAB. You should submit the `Loop.asm` and `Loop.lst` files.

Solution:

```
                      LIST  P=18F4520             ;directive to 
define processor
                      #include <P18F4520.INC>         ;processor 
specific variable definitions

;-----------------------------------------------------------------
-------------

                      CBLOCK 0x00
                      Sum
                      Qn1
                      Qn2
                      Qn3
                      Count
                      endc

MyN equ d'10'

;-----------------------------------------------------------------
                      ORG        0x0000
                      goto  Main ;go to start of main code
;----------------------------------------------------------------
-;Start of main program

Main:       movlw MyN-2
                movwf Count, A
                movlw d'1'
                movwf Qn1, A; this variable stores Q_{n-1}
                clrf Qn2, A
                clrf Qn3, A
                movlw d'1';initialize Sum as Q_0 + Q_1 + Q_2 = 1;
                movwf Sum, A


Loop:           movf Qn1, W, A
                addwf Qn2, W, A
                addwf Qn3, W, A;[WREG] = [Qn1] + [Qn2] + 
[Qn3] ;Note: WREG stores Qn
                addwf Sum, F, A;  [Sum] = [Sum] + [WREG]
                movff Qn2, Qn3
                movff Qn1, Qn2
                movwf Qn1, A

                decfsz Count, F, A
                bra Loop

                bra $
;*****************************************************************
;End of program
;
                      END
```

Question 2 (19 marks)

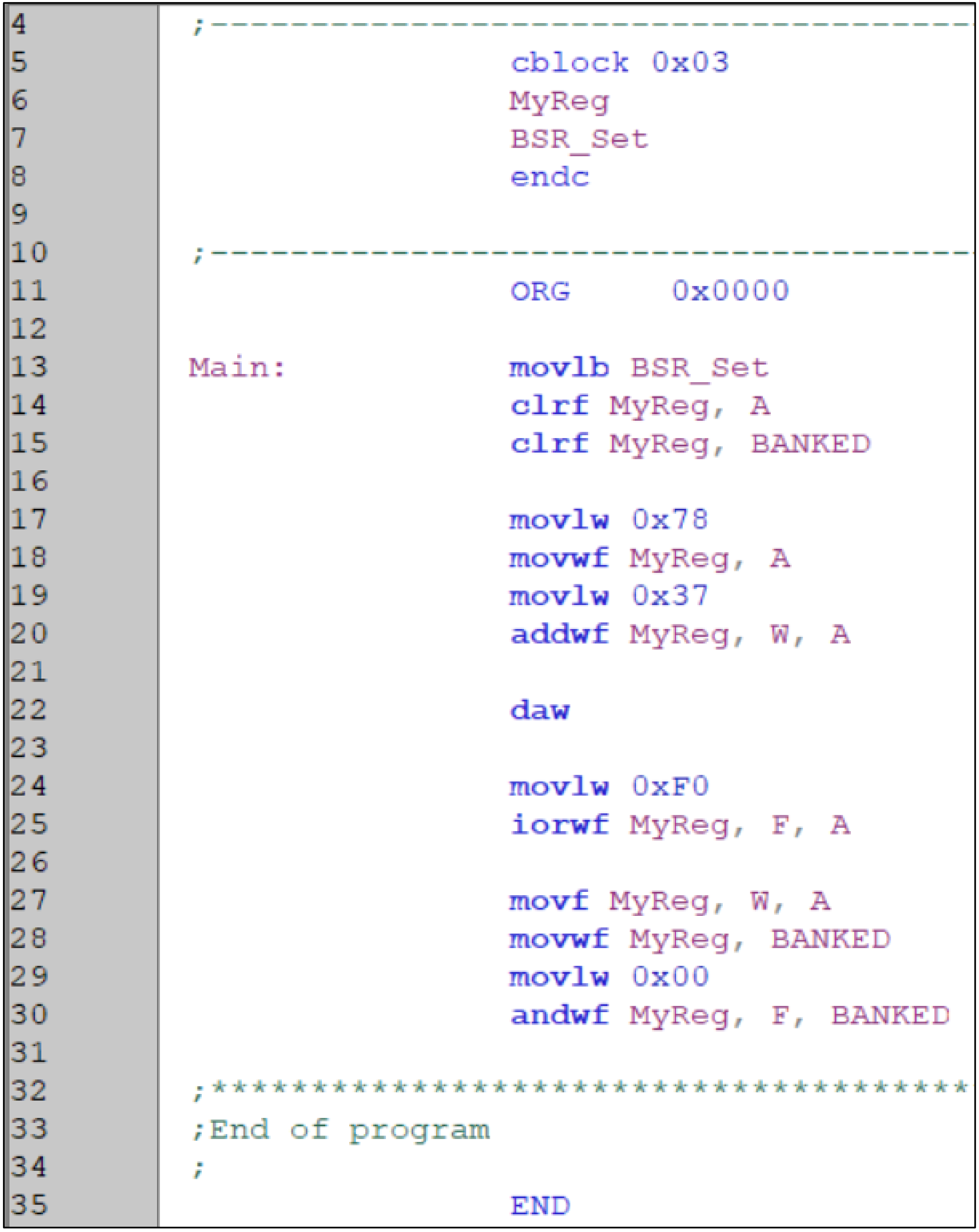
BCD addition of two mult-byte numbers

Write a program to add two decimal numbers 524198 and 487998. Note that the result should be stored in four file registers. Save your program in a file named `BCD.asm`. Assemble and verify your code using MPLAB. You should submit the `BCD.asm` and `BCD.lst` files.

```
                      LIST  P=18F4520;directive to define processor
                      #include <P18F4520.INC> ;processor specific
                                      ;variable definitions

                      CBLOCK 0x10
                      SumLow
                      SumHigh
                      SumUpper
                      SumUpper2
                      endc
;------------------------------------------------------------------
                      ORG        0x0000
                      goto  Main  ;go to start of main code
;------------------------------------------------------------------

Main:            movlw 0x99
                      addlw 0x99
                      daw
                      movwf SumLow, A

                      movlw 0x41
                      movwf SumHigh, A
                      movlw 0x99
                      addwfc SumHigh, W, A
                      daw
                      movwf SumHigh, A

                      movlw 0x52
                      movwf SumUpper, A
                      movlw 0x49
                      addwfc SumUpper, W, A
                      daw
                      movwf SumUpper, A

                      clrf SumUpper2, A
                      clrf WREG
                      addwfc SumUpper2, F, A

                      bra $

                      END
```

Question 3 (21 marks)
Use the program below to answer the following questions.

Line Number



```
4   ;--------------------------------------
5                   cblock 0x03
6                   MyReg
7                   BSR_Set
8                   endc
9
10  ;--------------------------------------
11                  ORG     0x0000
12
13  Main:           movlb BSR_Set
14                  clrf MyReg, A
15                  clrf MyReg, BANKED
16
17                  movlw 0x78
18                  movwf MyReg, A
19                  movlw 0x37
20                  addwf MyReg, W, A
21
22                  daw
23
24                  movlw 0xF0
25                  iorwf MyReg, F, A
26
27                  movf MyReg, W, A
28                  movwf MyReg, BANKED
29                  movlw 0x00
30                  andwf MyReg, F, BANKED
31
32  ;**************************************
33  ;End of program
34  ;
35                  END
```

(a) Write down the **12-bit address** of the memory location in which the result of the `iorwf` operation in Line 25 of the above program is stored.

(b) Write down the **12-bit address** of the memory location in which the result of the `andwf` operation in Line 30 of the above program is stored.

(c) What are the statuses of the five flags in the STATUS register immediately following the execution of **Lines 20, 25, 30**? *You must demonstrate how you come up with the statuses of the five flags to receive credit.*

(d) What is the value stored in WREG after the execution of Line 22? Determine the five flags in the STATUS register immediately after the execution of Line 22. *You must demonstrate how your answers to receive credit.*

Solution:

(a) 0x003
(b) 0x403

The order of the status flags in the STATUS register is:

0	0	0	N	OV	Z	DC	C

Here are the values of the STATUS register.

(c) Line 20: 0x18
Line 25: 0x19
Line 30: 0x0D

(d) WREG: 0x15, STATUS = 0x19.

Question 4 (30 marks)

(a) For the following program, calculate the relative/absolute addresses (marked by “??”). You must show detailed calculations. Otherwise, you will obtain 0 mark in this question.

(b) Provide a brief description of the goal of the program. (Hint: What is stored in FinalReg?)

(c) Analyze the source code, draw a chart describing the flow of the program and explain how the goal stated in your answer to Part (b) is accomplished.

```
Program
Memory   Machine        LINE                 SOURCE
Address  Code
                        00005                 CBLOCK 0x000
                        00006                 FirstReg
                        00007                 SecondReg
                        00008                 ThirdReg
                        00009                 FinalReg
                        00010                 endc
                        00011
000000                  00012                 org 0x000000
000000 EF?? F???        00013                 goto Main
                        00014 ;-----------------------------
000048                  00015                org 0x000048
000048 0E3C             00016 Main:          movlw d'60'
00004A 6E00             00017                movwf FirstReg, A
00004C 0E16             00018                movlw d'22'
00004E 6E01             00019                movwf SecondReg, A
000050 0E37             00020                movlw d'55'
000052 6E02             00021                movwf ThirdReg, A
                        00022
000054 5000             00023 Here:          movf FirstReg, W, A
000056 6E03             00024                movwf FinalReg, A
000058 5C01             00025                subwf SecondReg, W, A
00005A E6??             00026                bn Final2
00005C D???             00027                bra Continue
00005E C001 F003        00028 Final2:        movff SecondReg, FinalReg
000062 5003             00029 Continue:      movf FinalReg, W, A
000064 6002             00030                cpfslt ThirdReg, A
000066 D???             00031                bra Over
000068 C002 F003        00032 Final3:        movff ThirdReg, FinalReg
00006C D???             00033 Over:          bra $
```

Solution:

(a) 4 marks for each correct answers with explanation. 0 mark if answers are given without explanation. 16 marks in total.

```
                      00005                   CBLOCK 0x000
                      00006                   FirstReg
                      00007                   SecondReg
                      00008                   ThirdReg
                      00009                   FinalReg
                      00010                   endc
                      00011
000000                00012                   org 0x000000
000000 EF24 F000      00013                   goto Main
                      00014 ;-----------------------------
000048                00015                  org 0x000048
000048 0E3C           00016 Main:            movlw d'60'
00004A 6E00           00017                  movwf FirstReg, A
00004C 0E16           00018                  movlw d'22'
00004E 6E01           00019                  movwf SecondReg, A
000050 0E37           00020                  movlw d'55'
000052 6E02           00021                  movwf ThirdReg, A
                      00022
000054 5000           00023 Here:            movf FirstReg, W, A
000056 6E03           00024                  movwf FinalReg, A
000058 5C01           00025                  subwf SecondReg, W, A
00005A E601           00026                  bn Final2
00005C D002           00027                  bra Continue
00005E C001 F003      00028 Final2:          movff SecondReg,
FinalReg
000062 5003           00029 Continue:        movf FinalReg, W, A
000064 6002           00030                  cpfslt ThirdReg, A
000066 D002           00031                  bra Over
000068 C002 F003      00032 Final3:          movff ThirdReg,
FinalReg
00006C D7FF           00033 Over:            bra $
```

(b) The program compares the values stored in three registers: FirstReg, SecondReg and ThirdReg, and stores the minimum value in FinalReg.

(c) The flowchart should communicate the following three points.

The workflow is (1) Assume FirstReg contains the minimum value and copies over its value to FinReg; (2) Perform [SecondReg] – [FinalReg]. If the result is negative, [FinalReg] = [SecondReg]; (3) Compare the value stored in ThirdReg with FinalReg using cpfslt. [ThirdReg] < [FinalReg], replace the [FinalReg] by [ThirdReg].