T62 Tutorial 4

Enter the following program. **X** is 3 if the last digit of your student ID number is 1, 3, 5, 7, or 9. **X** is 4 if the last digit of your student ID number is 0, 2, 4, 6, or 8. **Y** is the last digit of your student ID number plus one. **Z** is the second last digit of your student ID number.

```
        LIST    P=18F4520
        #include <P18F4520.INC>

        cblock 0xX0
                mem1
                mem2:d'8'
                mem3:0e
                mem4
        endc

        ORG     0x0000
        goto    Main
        ORG     0x00Y0
Main:   movlw a'a'
        movwf mem1
        movlw d'Z0'
        movwf mem2
        movlw b'01101100'
        addwf  mem2,w
        movwf mem3
        movlw 0f
        andwf  mem3,w
        movwf mem4
Here:   goto    Here
        nop
        END
```

1. Copy the program from the list file.

(2 marks)

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001         LIST    P=18F4520
                     00002         #include <P18F4520.INC>
                     00001         LIST
                     00002
                     00003 ;==========================================================================
                     00004 ;  MPASM PIC18F4520 processor include
                     00005 ;
                     00006 ;  (c) Copyright 1999-2013 Microchip Technology, All rights reserved
                     00007 ;==========================================================================
                     00008
                     01488         LIST
                     00003
                     00004         cblock 0x40
  00000040           00005                 meml
  00000041           00006                 mem2:d'8'
  00000049           00007                 mem3:0e
  00000057           00008                 mem4
                     00009         endc
                     00010         ; student ID number is 12345678
000000               00011                 ORG     0x0000
000000 EF48 F000     00012                 goto    Main
000090               00013                 ORG     0x0090
000090 0E61          00014 Main:   movlw   a'a'
000092 6E40          00015         movwf   meml
000094 0E46          00016                 movlw   d'70'
000096 6E41          00017         movwf   mem2
000098 0E6C          00018                 movlw   b'01101100'
00009A 2441          00019                 addwf   mem2,w
00009C 6E49          00020                 movwf   mem3
00009E 0E0F          00021                 movlw   0f
0000A0 1449          00022                 andwf   mem3,w
0000A2 6E57          00023                 movwf   mem4
0000A4 EF52 F000     00024 Here:   goto    Here
0000A8 0000          00025         nop
                     00026                 END
```

2. Show the opcodes of movlw, movwf, addwf, andwf, and goto.

(2 marks)

movlw	0E
movwf	6E
addwf	24
andwf	14
goto	EFF

3. Show the operand of goto Here instruction in binary representation.

(2 marks)

0101 0010 0000 0000 0000

4. Show the calculation of PC for goto Here instruction in binary representation. Finally, show PC in hexadecimal representation.

(4 marks)

PC = 0000 0000 0000 0101 0010 + 0 = 0 0000 0000 0000 1010 0100
PC = 0000A4

5. What are the memory addresses of mem1, mem2, mem3, and mem4?

(2 marks)

mem1 0x040
mem2 0x041
mem3 0x049
mem4 0x057

6. After the program is executed, what are the contents of mem1, mem2, mem3, and mem4?

(2 marks)

mem1 0x61
mem2 0x46
mem3 0xB2
mem4 0x02

Enter the following program. **X** is the last digit of your student ID number. **Y** is the second last digit of your student ID number. **Z** is the third last digit of your student ID number.

```
        LIST    P=18F4520
        #include <P18F4520.INC>

        ORG     0x0000
        goto    Main
        ORG     0x0060
Main:   movlw   0x57
        addlw   0xXF
        movlw   0xAB
        andlw   0x5Y
        movlw   0x32
        xorlw   0xZD
Here:   goto    Here
        nop
        END
```

7. Copy the program from the list file.

(2 marks)

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001         LIST    P=18F4520
                     00002         #include <P18F4520.INC>
                     00001         LIST
                     00002
                     00003 ;=======================================================================
                     00004 ;  MPASM PIC18F4520 processor include
                     00005 ;
                     00006 ;  (c) Copyright 1999-2013 Microchip Technology, All rights reserved
                     00007 ;=======================================================================
                     00008
                     01488         LIST
                     00003         ; student ID number is 12345678
000000               00004                 ORG     0x0000
000000 EF30 F000     00005                 goto    Main
000060               00006                 ORG     0x0060
000060 0E57          00007 Main:   movlw   0x57
000062 0F8F          00008         addlw   0x8F
000064 0EAB          00009         movlw   0xAB
000066 0B57          00010                 andlw   0x57
000068 0E32          00011                 movlw   0x32
00006A 0A6D          00012                 xorlw   0x6D
00006C EF36 F000     00013 Here:   goto    Here
000070 0000          00014         nop
                     00015                 END
```

8. Show the contents of WREG and STATUS register after addlw 0x**X**F is executed. Explain why you observe the status of the flag bits.

(2 marks)

WREG = E6 STATUS = 12
DC = 1 because there is carry from bit 3 to bit 4
N = 1 because bit 7 of WREG is 1
C = 0 because no carry
Z = 0 because WREG is not zero
OV = 0 because +ve + -ve has no overflow problem

9. Show the contents of WREG and STATUS register after andlw 0x5**Y** is executed. Explain why you observe the status of the flag bits.

(2 marks)

WREG = 03 STATUS = 02
DC = 1 because andlw does not affect DC
N = 0 because bit 7 of WREG is 0
C = 0 because andlw does not affect C
Z = 0 because WREG is not zero
OV = 0 because andlw does not affect OV

10. Show the contents of WREG and STATUS register after xorlw 0x**Z**D is executed. Explain why you observe the status of the flag bits.

(2 marks)

WREG = 5F STATUS = 02
DC = 1 because xorlw does not affect DC
N = 0 because bit 7 of WREG is 0
C = 0 because xorlw does not affect C
Z = 0 because WREG is not zero
OV = 0 because xorlw does not affect OV

Enter the following program. **X** is the second last digit of your student ID number plus one. **Y** is the last digit of your student ID number plus one. Set the frequency to 4 MHz.

```
        LIST    P=18F4520
        #include <P18F4520.INC>

        MyReg   EQU   0x0F

          ORG   0x0000
          goto  Main

          ORG   0x0060
Main:   nop
          call  Delay
          nop
Here:   goto  Here
Delay:  movlw 0xXY
          movwf MyReg
Again:  nop
          nop
          nop
          nop
          decf  MyReg,F
          bnz   Again
          return
          END
```

11. Copy the program from the list file.

(2 marks)

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001         LIST    P=18F4520
                     00002         #include <P18F4520.INC>
                     00001         LIST
                     00002
                     00003 ;=======================================================================
                     00004 ;  MPASM PIC18F4520 processor include
                     00005 ;
                     00006 ;  (c) Copyright 1999-2013 Microchip Technology, All rights reserved
                     00007 ;=======================================================================
                     00008
                     01488         LIST
                     00003         ; student ID number is 12345678
Warning[207]: Found label after column 1. (MyReg)
  0000000F           00004         MyReg   EQU     0x0F
                     00005
000000               00006                 ORG     0x0000
000000 EF30 F000     00007                 goto    Main
                     00008
000060               00009                 ORG     0x0060
000060 0000          00010 Main:   nop
000062 EC36 F000     00011                 call    Delay
000066 0000          00012                 nop
000068 EF34 F000     00013 Here:   goto    Here
00006C 0E89          00014 Delay:  movlw   0x89
00006E 6E0F          00015                 movwf   MyReg
000070 0000          00016 Again:  nop
000072 0000          00017                 nop
000074 0000          00018                 nop
000076 0000          00019                 nop
000078 060F          00020                 decf    MyReg,F
00007A E1FA          00021                 bnz     Again
00007C 0012          00022                 return
                     00023                 END
```

12. Execute the program with “Step Into” button. Examine the Stopwatch window. Show the calculation to find the number of instruction cycles required by the Delay function.

(4 marks)

Number of instruction cycles = 2 + 1 + 1 + (137 x 7) – 1 + 2 = 964

13. Execute the program with “Step Over” button. Examine the Stopwatch window. What is the time delay generated by the Delay function.

(2 marks)

964 µsec