**Test**

1. Write a program to calculate the sum of four packed BCD numbers. The numbers are from your student ID number. For instance, if the ID is 12345678, the BCD numbers are 12, 34, 56, and 78. The sum result must be in packed BCD. Copy the program from the MPLAB editor window. Copy the sum result from the Watch window.

(4 marks)

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

ORG 0x0000

CBLOCK 0x0010

L\_Byte

H\_Byte

D1

D2

D3

D4

ENDC

goto Start

ORG 0x0080

Start: ;56 04 66 80

movlw 0x56

movwf D1

movlw 0x04

movwf D2

movlw 0x66

movwf D3

movlw 0x80

movwf D4

clrf L\_Byte

clrf H\_Byte

movf L\_Byte, w

addwf D1, w

daw

bnc N\_1

incf H\_Byte,f

N\_1:

addwf D2, w

daw

bnc N\_2

incf H\_Byte,f

N\_2:

addwf D3, w

daw

bnc N\_3

incf H\_Byte,f

N\_3:

addwf D4, w

daw

bnc N\_4

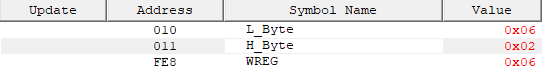
incf H\_Byte,f

N\_4:

movwf L\_Byte

Here: goto Here

END



Sum result = 206H

1. Perform arithmetic operation C = A – B, where A, B and C are 8-bit 2’s complement numbers. There are five numbers of A stored in locations **X**0h to **X**4h. Similarly the five numbers of B are stored in locations **Y**0h to **Y**4h. The results of C are saved in locations **Z**0h to **Z**4h. **X(1)** is the last digit of your student ID number plus one. **Y(9)** is the second last digit of your student ID number plus one. **Z(7)** is the third last digit of your student ID number plus one.
2. Write a program to perform the arithmetic operation using register indirect addressing mode without a loop.

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

ORG 0x0000

cblock 0x10

char\_A: 5

endc

cblock 0x90

char\_B: 5

endc

cblock 0x70

char\_C: 5

endc

lfsr 0, char\_A

lfsr 1, char\_B

lfsr 2, char\_C

movf INDF0, w

subwf INDF1, w

movwf INDF2

incf FSR0L, f

incf FSR1L, f

incf FSR2L, f

movf INDF0, w

subwf INDF1, w

movwf INDF2

incf FSR0L, f

incf FSR1L, f

incf FSR2L, f

movf INDF0, w

subwf INDF1, w

movwf INDF2

incf FSR0L, f

incf FSR1L, f

incf FSR2L, f

movf INDF0, w

subwf INDF1, w

movwf INDF2

incf FSR0L, f

incf FSR1L, f

incf FSR2L, f

movf INDF0, w

subwf INDF1, w

movwf INDF2

END

1. Write a program to perform the arithmetic operation using register indirect addressing mode with a loop.

(6 marks)

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

ORG 0x0000

count EQU 0x05

cblock 0x10

char\_A: 5

endc

cblock 0x90

char\_B: 5

endc

cblock 0x70

char\_C: 5

endc

lfsr 0, char\_A

lfsr 1, char\_B

lfsr 2, char\_C

movlw d'5'

movwf count

Loop:

movf INDF0, w

subwf INDF1, w

movwf INDF2

incf FSR0L, f

incf FSR1L, f

incf FSR2L, f

decf count, f

bnz Loop

END

1. PIC18 is running with a 16 MHz crystal oscillator. Write a program to perform the following tasks:
2. AN**X(0)** pin of PIC18 is connected to a potentiometer. **X(0)** is the last digit of your student ID number. The voltage range of the potentiometer is from 0V to 5V. The main program is to measure the voltage applied to the AN**X(0)** pin, convert it, and transfer the conversion result to Port D. Let the minimum value of TAD = 1.8 µsec and the minimum value of TACQ = 10 µsec.

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

CONFIG OSC = XT

CONFIG WDT = OFF

CONFIG LVP = OFF

ORG 0x0000

goto Start

ORG 40

Start:

movlw b'00000001' ; select channel AN0 and enable A/D

movwf ADCON0, A

movlw b'00000001' ; use VDD & VSS as reference voltages &

movwf ADCON1, A ; configure channel AN0 as analog input

movlw b'01010011' ; tad = (16e6/32)^-1 = 2e-6 > 1.8 tacq = 2x6

movwf ADCON2, A ; 01 010 011 select right justification

clrf TRISD

clrf PORTD

Here:

bsf ADCON0, GO, A ; start A/D conversion

wait\_con:

btfsc ADCON0, DONE, A ; wait until conversion is done

bra wait\_con

movff ADRESH, PORTD ; transfer the conversion result to Port D

goto Here

END

1. Use Timer0 interrupt to generate a 3 KHz square wave on pin RB6.

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

CONFIG OSC = XT

CONFIG WDT = OFF

CONFIG LVP = OFF

ORG 0x0000

bcf TRISB, 6

movlw 0x08

movwf T0CON ;cycle t = 0.25e-6

Here:

movlw 0xfd

movwf TMR0H

movlw 0x65

movwf TMR0L

bcf INTCON, TMR0IF

call Delay

btg PORTB, 6

bra Here

Delay:

bsf T0CON, TMR0ON

Again:

btfss INTCON,TMR0IF

bra Again

bcf T0CON, TMR0ON

return

END

1. Use Timer1 interrupt to generate a 30 Hz square wave on pin RB7.

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

CONFIG OSC = XT

CONFIG WDT = OFF

CONFIG LVP = OFF

ORG 0x0000

bcf TRISB, 7

movlw 0x08

movwf T1CON ;cycle t = 0.25e-6

Here:

movlw 0x00

movwf TMR1H

movlw 0x01

movwf TMR1L

bcf INTCON, TMR1IF

call Delay

btg PORTB, 7

bra Here

Delay:

bsf T1CON, TMR1ON

Again:

btfss INTCON,TMR1IF

bra Again

bcf T1CON, TMR1ON

return

END

(10 marks)

**Submission**

Enter your name, student ID number, and the answers in the MS Word document file. Re-name the file with your student ID number, e.g. 12345678.docx. **Deduct 4 marks for wrong file name.**

Submit the file by e-mail ([itklchan@cityu.edu.hk](mailto:itklchan@cityu.edu.hk)) before 9:00 pm. **Late submission will not be accepted.**