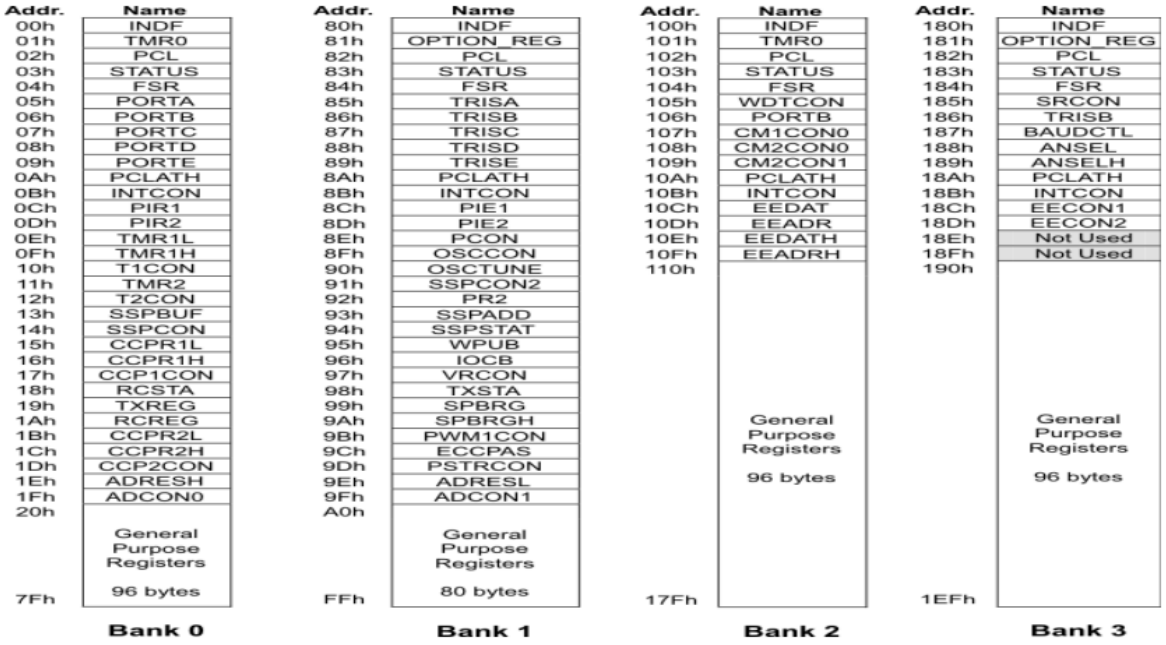
قائمة الاسئلة
الانظمة المدمجة ـ كلية الهندسة ـ قسم الميكاترونكس- المستوى الرابع ـ ساعتان ـ درجة هذا الاختبار (50)
د.فاروق الفهيدي

1)



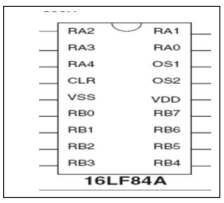
SFRs bank 0

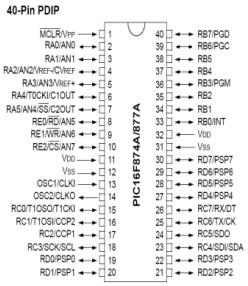
Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00h	INDF	Indirect register							
01h	TMR0	Timer T0 Register							
02h	PCL	Least Significant Byte of Program Counter							
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
04h	FSR	Indirect Data Memory Address Pointer							
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
09h	PORTE	-	-	-	-	RE3	RE2	RE1	RE0
0Ah	PCLATH	-	-	-	Upper 5 bits of Program Counter				
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0Dh	PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	-	CCP2IF
0Eh	TMR1L	Least Significant Byte of the 16-bit Timer TMR0							
0Fh	TMR1H	Most Significant Byte of the 16-bit Timer TMR0							
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
11h	TMR2	Timer T2 Register							
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register							
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
15h	CCPR1L	Capture/ComparePWM Register 1 Low Byte (LSB)							
16h	CCPR1H	Capture/ComparePWM Register 1 High Byte (LSB)							
17h	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
19h	TXREG	EUSART Transmit Data Register							
1Ah	RCREG	EUSART Receive Data Register							
1Bh	CCPR2L	Capture/Compare PWM Register 1 Low Byte (LSB)							
1Ch	CCPR2H	Capture/Compare PWM Register 1 High Byte (LSB)							
1Dh	CCP2CON	-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
1Eh	ADRESH	A/D Result Register High Byte							
1Fh	ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

SFRs bank 3

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
180h	INDF	Indirect Register							
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
182h	PCL	Least Significan Byte of the Program Counter							
183h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
184h	FSR	Indirect Data Memory Address Pointer							
185h	SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	-	FVREN
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
187h	BAUDCTL	ABDOVF	RCIDL	-	SCKP	BRG16	-	WUE	ABDEN
188h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
189h	ANSELH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
19Ah	PCLATH	-	-	-	Upper 5 bits of the Program Counter				
19Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
19Ch	EECON1	EEPGD	-	-	-	WRERR	WREN	WR	RD
19Dh	EECON2	EEPROM Control Register 2							

File Address			File Address
00h	Indirect addr.(1)	Indirect addr.(1)	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h





0000 = Capture/Compare/PWM disabled (resets CCPx module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCPxIF bit is set)
1001 = Compare mode, clear output on match (CCPxIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)
1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

CCPx Mode Select bit

Embedded Systems are used for:

1) - Computational
2) - General purposes
3) + Specific purposes
4) - all of these

2)

Addr.	Name
00h	INDF
01h	TMR0
02h	PCL
03h	STATUS
04h	FSR
05h	PORTA
06h	PORTB
07h	PORTC
08h	PORTD
09h	PORTE
0Ah	PCLATH
0Bh	INTCON
0Ch	PIR1
0Dh	PIR2
0Eh	TMR1L
0Fh	TMR1H
10h	T1CON
11h	TMR2
12h	T2CON
13h	SSPBUF
14h	SSPCON
15h	CCPR1L
16h	CCPR1H
17h	CCP1CON
18h	RCSTA
19h	TXREG
1Ah	RCREG
1Bh	CCPR2L
1Ch	CCPR2H
1Dh	CCP2CON
1Eh	ADRESH
1Fh	ADCON0
20h	General Purpose Registers 96 bytes
7Fh	

Bank 0

Addr.	Name
80h	INDF
81h	OPTION_REG
82h	PCL
83h	STATUS
84h	FSR
85h	TRISA
86h	TRISB
87h	TRISC
88h	TRISD
89h	TRISE
8Ah	PCLATH
8Bh	INTCON
8Ch	PIE1
8Dh	PIE2
8Eh	PCON
8Fh	OSCCON
90h	OSCTUNE
91h	SSPCON2
92h	PR2
93h	SSPADD
94h	SSPSTAT
95h	WPUB
96h	IOCB
97h	VRCON
98h	TXSTA
99h	SPBRG
9Ah	SPBRGH
9Bh	PWM1CON
9Ch	ECCPAS
9Dh	PSTRCON
9Eh	ADRESL
9Fh	ADCON1
A0h	General Purpose Registers 80 bytes
FFh	

Bank 1

Addr.	Name
100h	INDF
101h	TMR0
102h	PCL
103h	STATUS
104h	FSR
105h	WDTCON
106h	PORTB
107h	CM1CON0
108h	CM2CON0
109h	CM2CON1
10Ah	PCLATH
10Bh	INTCON
10Ch	EEDAT
10Dh	EEADR
10Eh	EEDATH
10Fh	EEADRH
110h	General Purpose Registers 96 bytes
17Fh	

Bank 2

Addr.	Name
180h	INDF
181h	OPTION_REG
182h	PCL
183h	STATUS
184h	FSR
185h	SRCON
186h	TRISB
187h	BAUDCTL
188h	ANSEL
189h	ANSELH
18Ah	PCLATH
18Bh	INTCON
18Ch	EECON1
18Dh	EECON2
18Eh	Not Used
18Fh	Not Used
190h	General Purpose Registers 96 bytes
1EFh	

Bank 3

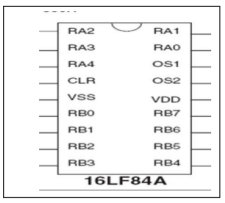
SFRs bank 0

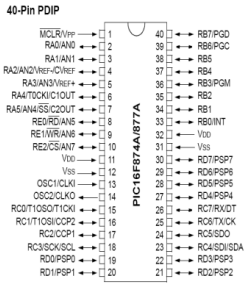
Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00h	INDF	Indirect register							
01h	TMR0	Timer T0 Register							
02h	PCL	Least Significant Byte of Program Counter							
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
04h	FSR	Indirect Data Memory Address Pointer							
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
09h	PORTE	-	-	-	-	RE3	RE2	RE1	RE0
0Ah	PCLATH	-	-	-	Upper 5 bits of Program Counter				
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0Dh	PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	-	CCP2IF
0Eh	TMR1L	Least Significant Byte of the 16-bit Timer TMR0							
0Fh	TMR1H	Most Significant Byte of the 16-bit Timer TMR0							
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
11h	TMR2	Timer T2 Register							
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register							
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
15h	CCPR1L	Capture/ComparePWM Register 1 Low Byte (LSB)							
16h	CCPR1H	Capture/ComparePWM Register 1 High Byte (LSB)							
17h	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
19h	TXREG	EUSART Transmit Data Register							
1Ah	RCREG	EUSART Receive Data Register							
1Bh	CCPR2L	Capture/Compare PWM Register 1 Low Byte (LSB)							
1Ch	CCPR2H	Capture/Compare PWM Register 1 High Byte (LSB)							
1Dh	CCP2CON	-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
1Eh	ADRESH	A/D Result Register High Byte							
1Fh	ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

SFRs bank 3

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
180h	INDF	Indirect Register							
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
182h	PCL	Least Significan Byte of the Program Counter							
183h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
184h	FSR	Indirect Data Memory Address Pointer							
185h	SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	-	FVREN
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
187h	BAUDCTL	ABDOVF	RCIDL	-	SCKP	BRG16	-	WUE	ABDEN
188h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
189h	ANSELH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
19Ah	PCLATH	-	-	-	Upper 5 bits of the Program Counter				
19Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
19Ch	EECON1	EEPGD	-	-	-	WRERR	WREN	WR	RD
19Dh	EECON2	EEPROM Control Register 2							

File Address			File Address
00h	Indirect addr.(1)	Indirect addr.(1)	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h





0000 = Capture/Compare/PWM disabled (resets CCPx module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCPxIF bit is set)
1001 = Compare mode, clear output on match (CCPxIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)
1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

CCPx Mode Select bit

Most microcontrollers have ……. architecture:

1) - Von-Neuman architecture
2) + Harvard Architecture
3) - both Von-Neuman & Harvard Architectures
4) - none of the them

3) All PIC16F's Microcontrollers are established with ……. (CPU) unit:
   1) + 8-bit Microprocessor
   2) - 8-bit Microcontroller
   3) - 16-bit Microprocessor
   4) - 16-bit Microcontroller
4) RISC architecture is characterized by:
   1) - the usage of large number of registers and limited access to Memory
   2) - Fixed Instructions & Registers size
   3) + All answers
   4) - none of these

5) PIC16F877 & PIC16F84 microcontrollers need _____ clock cycles to execute most of their Assembly instructions.
   1) - Two instruction cycle
   2) + 4
   3) - 2
   4) - 2, 4, or 8
6) The depth of the PIC16F877's stack has _____ levels of 13-bit width.
   1) - 2
   2) - 4
   3) - 6
   4) + 8
7) The Flash ROM can be used for:
   1) - Storing temporary data
   2) - storing of permanent data only
   3) + hosting the driving program & some of data variables like lookup
   4) - hosting the driving program only
8) The interrupt of the timer2 couldn't be used for waking up the PIC16F873-Mc when it is in a sleep mode.
   1) + true
   2) - false
9) The Timer1 of the PIC16F84-Mc has an internal register of 16-bit (Timer1L & Timer1H):
   1) - true
   2) + false
10) It's not true that any external interrupt can wake up the microcontroller from the sleep mode:
   1) - true
   2) + false
11) Some applications of the CCP module in PWM mode are:
   1) - Generating repetitive pulses, Measuring Frequency, DAC & Wake up MC
   2) + Generating repetitive pulses, Measuring Frequency & DAC
   3) - Generating repetitive pulses & Measuring Frequency
   4) - all previous answers are right, in addition to get the MC in sleep mode
12) The slowest clock tick interrupt rate, which can be generated using Timer0 without pre-scaler, and it operated with oscillator frequency of 4 MHz, is:
   1) - 15.62587 Hz
   2) + 3.906 KHz
   3) - 15.26 Hz
   4) - 3.25 KHz
13) The fastest clock tick interrupt rate, which can be generated using Timer0 operated with oscillator frequency of 4 MHz, is:
   1) + 1 MHz
   2) - 3.906 kHz
   3) - 4 MHz
   4) - none of the above
14) The slowest interrupt frequency, which can be generated using Timer2 operated with oscillator frequency of 4 MHz, is:
   1) - 15.627 Hz
   2) + 15.26 Hz
   3) - 15.25 Hz
   4) - 15.255 Hz
15) In order to employ Timer0 in counter mode for counting up to 1000 events, you should function/configure

its pre-scaler with at least _____.

1) - 256 Pre-Scale value
2) - 64 Pre-Scale value
3) - 16 Pre-Scale value
4) + 4 Pre-Scale value

16) Timer2 module on the PIC16F877-MC couldn't generate an interrupt.

1) - True
2) + False

17) The Assembly instruction, MOVLW 583, didn't affect any bit in the status register.

1) - True
2) + False

18) The Assembly instruction, MOVF Cntr, will affect (may change) the logic value of at least one bit in the status register.

1) - True
2) + False

19) After the execution of the Assembly instruction, ADDLW 0xFA, the bit value of the carry-Flag (CF) will be '1'. Assume, previous value of W-reg was 0x90.

1) + True
2) - False

20) Study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right):

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
   call Leftmot1
   call Rightmot1
   call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
   call Leftmot2
   call Rightmot2
   call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
GOTO LB1
; .........
Leftmot1
    bsf porta, mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
```

```
.
Leftmot2
    bsf porta,mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf porta, mot_en_right    ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf porta,mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

Note: this program may be repeated with different question
According to this driving program, the Derbot will move 1-minute to right and 1-minute to left and this process is repeated continuously.

1) - True
2) + False

21) Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right):

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
   call Leftmot1
   call Rightmot1
   call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
   call Leftmot2
   call Rightmot2
   call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
GOTO LB1
; .........
Leftmot1
    bsf porta, mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
```

```
    return
Leftmot2
    bsf porta,mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf porta, mot_en_right    ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf porta,mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

e: this program may be repeated with different question
cording to this driving program, the Derbot will move 1-minute to right and 1-minute to left and this process is repeated continuously until battery energy is decreased below the threshold level.

1) - True
2) + False

22)

Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right):

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
  call Leftmot1
  call Rightmot1
  call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
  call Leftmot2
  call Rightmot2
  call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
GOTO LB1
; .........
Leftmot1
    bsf  porta, mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
Leftmot2
    bsf  porta,mot_en_left       ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf  porta, mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf  porta,mot_en_right      ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

According to this program, neglect any other design considerations, the 2-motor forward and backward speeds are:

1) - Little bit Similar
2) - Different
3) - Approximately Equal
4) + Exactly Equal

23) Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right):

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
  call Leftmot1
  call Rightmot1
  call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
  call Leftmot2
  call Rightmot2
  call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
GOTO LB1
; .........
Leftmot1
    bsf  porta, mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
Leftmot2
    bsf  porta,mot_en_left       ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf  porta, mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf  porta,mot_en_right      ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

According to this program, neglect any other design considerations, and after 8-minutes the Derbot Position is at:

1) - The Origin Start Point
2) + The Destined/Designed Point
3) - The Middle point between the Start & the Designed Points
4) - None of the previous answers

24) Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right):

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
  call Leftmot1
  call Rightmot1
  call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
  call Leftmot2
  call Rightmot2
  call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
GOTO LB1
; .........
Leftmot1
    bsf  porta, mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
Leftmot2
    bsf  porta,mot_en_left       ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf  porta, mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf  porta,mot_en_right      ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

According to this program, neglect any other design considerations, and after 5-minutes the Derbot Position is at:

1) - The Origin Start Point

2) + The Destined/Designed Point

□□□

3) - The Middle point between the Start & the Designed Points

4) - None of the previous answers

25) Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right) with bit little modification in main proc:

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
    call Leftmot1
    call Rightmot1
    call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
    call Leftmot1
    bcf  porta, mot_en_right     ; L293D Right H-bridge Enable bit
; Assume the following subroutine generates a delay time of (4 sec) enough for 90° Rotation
    call Delay_Sub2
GOTO LB1
```

```
; .....
Leftmot1
    bsf  porta, mot_en_left     ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
Leftmot2
    bsf  porta,mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf  porta, mot_en_right    ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf  porta,mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

According to this program situation, neglect any other design considerations, the Derbot is driven in:

1) - A circular fashion starting from and passing through the Starting Point

2) - A spherical fashion starting from and passing through the Starting Point

3) - A Rectangular way starting from and passing through the Starting Point

4) + A Squaral round starting from and passing through the Starting Point

26) Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right) with bit little modification in main proc:

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
    call Leftmot1
    call Rightmot1
    call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
    call Leftmot1
    bcf  porta, mot_en_right     ; L293D Right H-bridge Enable bit
; Assume the following subroutine generates a delay time of (4 sec) enough for 90° Rotation
    call Delay_Sub2
```

```
; .....
Leftmot1
    bsf  porta, mot_en_left     ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
Leftmot2
    bsf  porta,mot_en_left      ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf  porta, mot_en_right    ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf  porta,mot_en_right     ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

According to this program, neglect any other design considerations, and at first-time execution of the GOTO LB1 instruction, the Derbot Position is at:

1) - The initial point of the second quarter arc of the circular round

2) - The initial point of the second quarter arc of the Spherical round

3) - The initial point of the second side (assume width) of the rectangular round

4) + The initial point of the second side of the Squaral round

27)

Again, study the following assembly code (program) for driving a Derbot with 2-motors (Left & Right) *with bit little modification in main proc*:

```
; Initiallization & set up of PWM
    movlw B'00000100'
    movwf t2con
    movlw B'00001100'
    movwf ccp1con
    movwf ccp2con
    movlw 0f9
    movwf pr2
Main ; Main Procedure (Subroutine) Start Point
LB1
   call Leftmot1
   call Rightmot1
   call Delay_Sub; Assume this subroutine generates a delay time of 1-minute
   call Leftmot1
   bcf  porta, mot_en_right      ; L293D Right H-bridge Enable bit
; Assume the following subroutine generates a delay time of (4 sec) enough for 90° Rotation
    call Delay_Sub2
GOTO LB1
; .........
Leftmot1
    bsf  porta, mot_en_left     ; L293D Left H-bridge Enable bit
    movlw D'176'
    movwf CCPR2L
    return
Leftmot2
    bsf  porta, mot_en_left     ; L293D Left H-bridge Enable bit
    movlw D'74'
    movwf CCPR2L
    return
Rightmot1
    bsf  porta, mot_en_right    ; L293D Right H-bridge Enable bit
    movlw D'176'
    movwf CCPR1L
    return
Rightmot2
    bsf  porta, mot_en_right    ; L293D Right H-bridge Enable bit
    movlw D'74'
    movwf CCPR1L
    return
```

According to this program, neglect any other design considerations, the Derbot will back visit or reach (first touch) to its starting point after:

1) - 4 minutes
2) + 4 minutes and 12 seconds
3) - 4 minutes and 16 seconds
4) - Will not exactly back or pass throughout its original/starting point

28) According to the usage of the CCP module in PWM mode for Digital-to-Analog (DAC) conversion, an external _____ is needed.
1) + RC Filter circuit.
2) - Low-pass RC Filter circuit followed by bandpass filter.
3) - High-pass RC Filter circuit followed by bandpass filter.
4) - Band-pass RC Filter circuit followed by bandpass filter.

29) A Microcontroller like PIC16F84 which is developed without CCP on-chip module couldn't be used for generating a digital waveform.
1) - True
2) + False

30) A square waveform with 50% duty cycle which is generated using the CCP module in PWM mode, can drive a motor with:
1) - Suitable average current for guiding a motor in stable driving
2) - some +ve average current but, it doesn't enough to drive a motor
3) + a zero average current which stops the movement of a motor
4) - none of the above answers.

31) **Read and study the following assembly code for generating a software delay time.**
Assume a PIC16F873-MC operated with oscillator frequency of *4 MHz*.

```
SwDly
    movlw  0xf9          ; 0f9 is a Hexadecimal value
    movwf  cntr1
del
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    nop                  ;1T
    decfsz cntr1,1       ;1T
    goto   del           ;2T
    return
```

According to the above subroutine, assume there is another subroutine, let's name it OuterSub, which is used to invoke/call the SwDly subroutine using a loop with 200-repitions. Hence, the approximately generated delay time after OuterSub subroutine execution is equal to:

1) - 200 seconds
2) - 200 minutes
3) + 1 seconds
4) - 1 minute

32) **Read and study the following assembly code for generating a software delay time.**
Assume a PIC16F873-MC operated with oscillator frequency of *4 MHz*.

```
SwDly
    movlw  0f9          ; 0f9 is a Hexadecimal value
    movwf  cntr1
del
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    nop                 ;1T
    decfsz cntr1,1      ;1T
    goto   del          ;2T
    return
```

According to the above subroutine, assume there is another subroutine, let's name it OuterSub, which is used to invoke/call the SwDly subroutine using a loop with 200-repitions. Hence, how many calls are required from a third subroutine, named OuterSub2, to the OuterSub for generating a delay time of 4 minutes?

1) - Approximately 1 call, which generates 3 minutes and 20 seconds
2) + 240 calls
3) - Doesn't possible since the OuterSub is generating more than 4 minutes in 1-call
4) - 4 calls

33) **Read and study the following lookup subroutine:**

```
;   A lookup Subroutine named LkTbl
LkTbl
        addwfpcl
        retlw 23
        retlw 3f        ; look to the following 6 retlw.. instructions after this one
        retlw 47
        retlw 7f
        retlw a2
        retlw 1f
        retlw 03
        retlw 67
```

According to the above subroutine, what are the work register's output values when the input values given by caller subroutine are: 0, 4, and 8 in one-to-one?

a) NaN; a2; 67
b) 23; 4; NaN
c) 23; a2; NaN
d) None of these answers

Note that: The NaN Symbol means Unknown value.

1) - a
2) - b
3) + c
4) - d

34) Which is the most basic non-volatile memory?
1) - Flash memory
2) - PROM
3) - EPROM
4) + ROM

35) Which of the following memories has more speed in accessing data?
1) + SRAM
2) - DRAM
3) - EPROM
4) - EEPROM

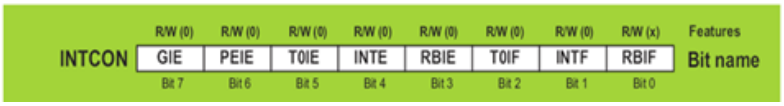
36) What is the purpose of address bus?
1) - to provide data to and from the chip
2) - to select a specified chip
3) + to select a location within the memory chip
4) - to select a read/write cycle View Answer

37) Which of the following is not a serial protocol?

1) - SPI
2) - I2C
3) - Serial port
4) + RS232

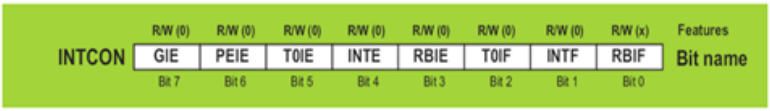
38) Which of the following is an ideal interface for LCD controllers?
1) - SPI
2) - parallel port
3) - Serial port
4) + Motorola-Bus

39) **At instance time while the program is executing. It is noticed that the *INTCON* register's value is B8H. Which type of interrupt(s) is (are) enabled?**



1) - No one interrupt is enabled
2) - Global Interrupt Enable bit only
3) + Timer0, External, and Port-B on-change interrupts
4) - All available interrupts are enabled

40) **During the work of the microcontroller, an interrupt is occurred, and the *INTCON* register's value is changed from *F8H* to *FAH*. Which interrupt source flag bit has been so-triggered?**



a) No one interrupt Flag bit is triggered
b) External Interrupt
c) Port-B on change Interrupt
d) Answers in (b) & (c)

1) - a
2) + b
3) - c
4) - d

41) **During the work of the microcontroller, an interrupt Service routine is now to be invoked & executed, and the *INTCON* register's value is found with *FBH* value. Which interrupt type is the source of next invocation, since the external interrupt service routine is in executing of its previous instruction of the its last instruction?**

1) - The External Interrupt
2) - Both the Port-B on change & the External Interrupt
3) - Timer0 Interrupt
4) + The Port-B on change Interrupt

42)

An interrupt Service routine is in execution, and the *INTCON* register's value is found with *F8H* value. Which interrupt type is the source?

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (x)	Features
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

1) - No Interrupt is happened
2) - The Port-B on change or the External Interrupt
3) - The Port-B on change, the External Interrupt or the Timer0 Interrupt
4) + unknown