National University of Computer and Emerging Sciences,
Lahore Campus

0022

Microprocessor Interfacing and Programming (EE3002)
Date: December 16th 2024
Course Instructor(s)
Mr. Maaz Rizvi

Final Exam
Total Time (Hrs): 3
Total Marks: **50**
Total Questions: **5**

Roll No _____ Section A _____ Student Signature _____

Do not write below this line

Instructions

- **Attempt all questions**
- **Answer all questions on the separate answer sheet**
- **There are 2 pages of the question paper. Please turn over to see the rest of the questions.**
- **Make sure your code is neatly written. Excessive cutting/overwriting will be penalized.**

Question No. 1 (CLO No. 3) **Marks: 3+7**

(A) **Write** three key differences between a microprocessor and a microcontroller. Which one is more suitable for Embedded applications and why?

(B) **Draw** the memory map to interface 64 KB of RAM and 256 KB of EPROM with 8086 processor. Also include the number of data lines, and address lines for each RAM and EPROM.

Question No. 2 (CLO No. 4) **Marks: 10**

Compose a code in PIC Assembly Language for the following task

The CPU continuously performs the task of transferring data from PORTC to PORTD. After the switch SW1 is pressed, the buzzer will start beeping with a frequency of 0.2 kHz. The buzzer will only go off after completing 300 beep cycles or when SW2 is pressed. (You must show your calculations for the timers)

Question No. 3 (CLO No. 4) **Marks: 3+5**

(A) **Formulate** the pros and cons of using C and Assembly Language to program microcontrollers.

(B) **Compose** a program in C18 that checks the combination of RB0, RB1, and RB2 and performs the operation on A and B according to the following table

RB2	RB1	RB0	OPERATION
0	0	0	**ADD** A AND B
0	0	1	**SUBTRACT** A AND B
0	1	0	**MULTIPLY** A AND B
0	1	1	**DIVIDE** A by B
1	0	0	**INVERT** A
1	0	1	**XOR** A AND B
1	1	0	**AND** A AND B
1	1	1	**OR** A AND B

Question No. 4 (CLO No. 4) **Marks: 10**

Construct a code in PIC Assembly Language for the following problem statement: The CPU constantly transfers data from PORTC to PORTD. A switch is connected to External Interrupt pin to toggle LED 1. Timer0 and Timer 1 are used to toggle LED 2 and LED3 after 10us and 20us respectively. The priority order is as follows

Timer1 >> External Interrupt >> Timer0

Question No. 5 (CLO No. 4) **Marks: 4+8**

(A) **Construct** a code in PIC Assembly Language to toggle PORTD 700 times.

(B) **Create** a traffic light system in which RED is on for 5m sec, Yellow is on for 2m sec and Green is on for 5m sec. There is a pedestrian button which when pressed will make the red light freeze for 10m sec and then the normal sequence continues for the traffic light.

EEPGD	CFGS	--	FREE	WRERR	WERN	WR#	RD#

EECON1 (EEPROM Control Register

Interrupt	Flag bit (Register)	Enable bit (Register)	Priority (Register)
Timer0	TMR0IF (INTCON)	TMR0IE (INTCON)	TMR0IP (INTCON2)
Timer1	TMR1IF (PIR1)	TMR1IE (PIE1)	TMR1IP (IPR1)
Timer2	TMR2IF (PIR1)	TMR2IE (PIE1)	TMR2IP (IPR1)
Timer3	TMR3IF (PIR3)	TMR3IE (PIE2)	TMR3IP (IPR2)
INT1	INT1IF (PIR1)	INT1IE (PIE1)	INT1IP (INTCON3)
INT2	INT2IF (PIR1)	INT2IE (PIE1)	INT2IP (INTCON)
TXIF	TXIF (PIR1)	TXIE (PIE1)	TXIP (IPR1)
RCIF	RCIF (PIR1)	RCIE (PIE1)	RCIP (IPR1)
RB_INT	RBIF (INTCON)	RBIE (INTCON)	RBIP (INTCON2)

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

RCON register bits

INTCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

	INTEDG0	INTEDG1	INTEDG2				

INTCON2 Register

RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

RD16 D7 16-bit read/write enable bit
1 = Timer1 16-bit is accessible in one 16-bit operation
0 = Timer1 16-bit is accessible in two 8-bit operations

D6 Not used

T1CKPS2:T1CKPS0 D5 D4 Timer1 prescaler selector
0 0 = 1:1 Prescale value
0 1 = 1:2 Prescale value
1 0 = 1:4 Prescale value
1 1 = 1:8 Prescale value

T1OSCEN D3 Timer1 oscillator enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shutoff

T1SYNC D2 Timer1 synchronization (used only when TMR1CS = 1 for counter mode to synchronize external clock input)
If TMR1CS = 0 this bit is not used.

TMR1CS D1 Timer1 clock source select bit
1 = External clock from pin RC0/T1CKI
0 = Internal clock (Fosc/4 from XTAL)

TMR1ON D0 Timer1 ON and OFF control bit
1 = Enable (start) Timer1
0 = Stop Timer1

Figure 9-10. T1CON (Timer 1 Control) Register

TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

T5$R0ON D7 Timer0 ON and OFF control bit
1 = Enable (start) Timer0
0 = Stop Timer0

T08BIT D6 Timer0 8-bit/16-bit selector bit
1 = Timer0 is configured as an 8-bit timer/counter.
0 = Timer0 is configured as a 16-bit timer/counter.

T0CS D5 Timer0 clock source select bit
1 = External clock from RA4/T0CKI pin
0 = Internal clock (Fosc/4 from XTAL oscillator)

T0SE D4 Timer0 source edge select bit
1 = Increment on H-to-L transition on T0CKI pin
0 = Increment on L-to-H transition on T0CKI pin

PSA D3 Timer0 prescaler assignment bit
1 = Timer0 clock input bypasses prescaler.
0 = Timer0 clock input comes from prescaler output.

T0PS2:T0PS0 D2:D0 Timer0 prescaler selector
0 0 0 = 1:2 Prescale value (Fosc / 4 / 2)
0 0 1 = 1:4 Prescale value (Fosc / 4 / 4)
0 1 0 = 1:8 Prescale value (Fosc / 4 / 8)
0 1 1 = 1:16 Prescale value (Fosc / 4 / 16)
1 0 0 = 1:32 Prescale value (Fosc / 4 / 32)
1 0 1 = 1:64 Prescale value (Fosc / 4 / 64)
1 1 0 = 1:128 Prescale value (Fosc / 4 / 128)
1 1 1 = 1:256 Prescale value (Fosc / 4 / 256)

Figure 9-2. T0CON (Timer0 Control) Register

Interrupt	Flag bit	Register	Enable bit	Register
Timer0	TMR0IF	INTCON	TMR0IE	INTCON
Timer1	TMR1IF	PIR1	TMR1IE	PIE1
Timer2	TMR2IF	PIR1	TMR2IE	PIE1
Timer3	TMR3IF	PIR3	TMR3IE	PIE2
INT0 (RB0)	INT0IF	INTCON	INT0IE	INTCON
INT1 (RB1)	INT1IF	INTCON3	INT1IE	INTCON3
INT2 (RB2)	INT2IF	INTCON3	INT2IE	INTCON3
TXIF (Transmit)	TXIF	PIR1	TXIE	PIE1
RCIF (Receive)	RCIF	PIR1	RCIE	PIE1
High Priority	0008H			
Low Priority	0018H			

- ADDLW Literal
- ANDLW Literal
- SUBLW Literal
- XORLW Literal
- MULLW Literal
- ADDWF fileReg, d
- ADDWFC fileReg, d
- ANDWF fileReg, d
- IORWF fileReg, d
- SUBWFB fileReg, d
- SUBWF fileReg, d
- SUBWFB fileReg, d
- XORWF fileReg, d
- COMF fileReg, d
- DECF fileReg, d
- DECFSZ fileReg, d
- DECFSNZ fileReg, d
- INCF fileReg, d
- INCFSZ fileReg, d
- INCSNZ fileReg, d
- MOVF fileReg, d
- NEGF fileReg, d
- RLCF fileReg, d
- RINCF fileReg, d
- RRCF fileReg, d
- RRNCF fileReg, d
- SWAPF fileReg, d
- VAR_NAME EQU Literal/Location
- ORG
- BTG fileReg, bit
- BTSC fileReg, bit
- BSF fileReg, bit
- BCF fileReg, bit
- MOVFF fileReg1, fileReg2
- GOTO addr
- BTFSC fileReg, bit
- BTFSS fileReg, bit
- CALL addr
- RCALL addr
- RETURN
- RETFIE
- NOP
- CLRF fileReg
- CLRW
- SETF fileReg
- STATUS
- C - Carry flag
- DC - Digital Carry flag
- Z - Zero flag
- OV - Overflow flag
- N - Negative flag
- CPFSGT fileReg, d
- CPFSEQ fileReg, d
- CPFSLT fileReg, d
- BCF T0CON, TMR0ON
- BSF T0CON, TMR0ON
- INTCON - GIEL
- BSF Reg, TMR0ON
- BCF Reg, TMR0ON
- BSF Reg, TMR1ON
- BCF Reg, TMR1ON
-

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F