Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

DEPARTMENT OF MECHATRONICS ENGINEERING

III SEMESTER B.TECH. (MECHATRONICS)

END SEMESTER EXAMINATIONS, July 2021

SUBJECT: MICROCONTROLLER BASED SYSTEM DESIGN [MTE 2153]

Time: 2 Hours **MAX. MARKS: 40**

Instructions to Candidates:

- Answer **any four full** the questions.
- Any data not provided may suitably assumed. Data sheet is attached at the end.

Q. No	Question	M	CO	PO	LO	BL
1A.	Mention role of flags in B<cond> for signed and unsigned arithmetic operations.	**4**	**1**	**1**	**1**	**2**
1B.	Write an ARM assembly language program to compute the $3x^2+2x+6$ of 10 8-bit numbers stored in the location 0x000001B0 onwards. Store the result in the RAM location starting from 0x20000000.	**6**	**1, 4**	**6**	**5**	**3**
2A.	50 hexadecimal 32-bit numbers are written to the RAM of the MSP432 microcontroller. The starting address is 0x20000000. Write an assembly program for the following operations. a. Your code should scan the memory location of all 50 numbers. If a memory entry has a value between 0x00000080 and 0x00001FFF, then save it starting from the memory location 0x20000100. b. Scan the saved numbers. If the value of a memory entry is odd, do nothing. If the value is even, divide it by two iteratively until the result becomes odd or zero. Save the result to the same memory address.	**6**	**2**	**1, 2**	**1, 2**	**3**
2B.	Explain the difference between maskable and non-maskable interrupts with an example.	**4**	**3**	**1, 2**	**1, 2**	**2**
3A.	Write a C program to count the difference between the number of times the buttons S1 and S2, connected to pins P1.1 and P1.4 on the MSP432 Launch Pad, are pressed. a. The button pressing operation should be defined in an ISR. b. Observe the count value from the watch window. c. If the button S1 is pressed more than the button S2, then turn on the red color of LED2 connected to pin P2.0 on the MSP432 Launch Pad. Otherwise, turn on the green color of LED2 connected to pin P2.1 on the MSP432 Launch Pad. If the button press numbers are equal, then turn on the blue color of LED connected to pin P2.2 on the MSP432 LaunchPad.	**6**	**3**	**1, 2**	**1, 2**	**3**

3B.	Write the difference between pull up and pull-down resistor configuration for input pins.	**2**	**3**	**1, 2**	**1, 2**	**2**
4A.	Explain the functioning of watch dog timer if microcontroller falls in continuous loop.	**4**	**3**	**1, 2**	**1, 2**	**2**
4B.	MSP432P401R is used by a company to develop an Induction heater. Develop an Embedded C code to address the manufacturers specifications. Use Timer 32 module for this operation. Condition 1. Power ON and OFF is connected to P2.4 2. Assume it has 2 switches for manual control which are connected to P3.4 and P3.5. 4. If the input pin P3.4 is pressed, it switches on the heater connected to P4.2 for 10 seconds. 5. If the input pin P3.5 is pressed, it switches on the heater connected to P4.2 for 5 seconds.	**6**	**4**	**1, 2**	**1, 2**	**3**
5A.	Establish communication for sending the information say “g” to glow the LED of another board, and “s” to stop the glowing. Use two MSP432 Launchpad boards. The first MSP432 Launchpad will be used for the button. The second will be used for the LEDs. Establish a digital communication link between these two boards using UART communication mode. Use the respective cheat sheet given below.	**6**	**3**	**1, 2**	**1, 2**	**3**
5B.	Explain the difference between synchronous and asynchronous serial data communication between two devices.	**4**	**3**	**1, 2**	**1, 2**	**2**
6A.	Write an Embedded C code for traffic signal management for 4 crossroad system. Each road has red, yellow, and green signals connected to: P1.0, P1.1, P1.2 Street 1 P2.0, P2.1, P2.2 Street 2 P3.0, P3.1, P3.2 Street 3 P4.0, P4.1, P4.2 Street 4 on MSP432P401R. Design a system for traffic management system to control the smooth flow of vehicles using flowchart.	**10**	**4**	**1, 2**	**1, 2**	**6**

Data sheet:

TIMER 32 REGISTERS

Bits	7	6	5	3-2	1	0
TIMER32_y->CONTROL	**ENABLE**	**MODE**	**IE**	**PRESCALE**	**SIZE**	**ONESHOT**
TIMER32_CONTROL_**XX**_x	**1-enable,** **0-disable**	**1- free running mode,** **0- periodic**	**1-enable interrupt** **0-disable interrupt**	**0=/1, 1=/16, 2=/256**	**1- 32 bits,** **0-16 bits**	**1-oneshot** **0-wrap**

25	Timer32_INT1	T32_INT1_IRQn	T32_INT1_IRQHandler
26	Timer32_INT2	T32_INT2_IRQn	T32_INT2_IRQHandler

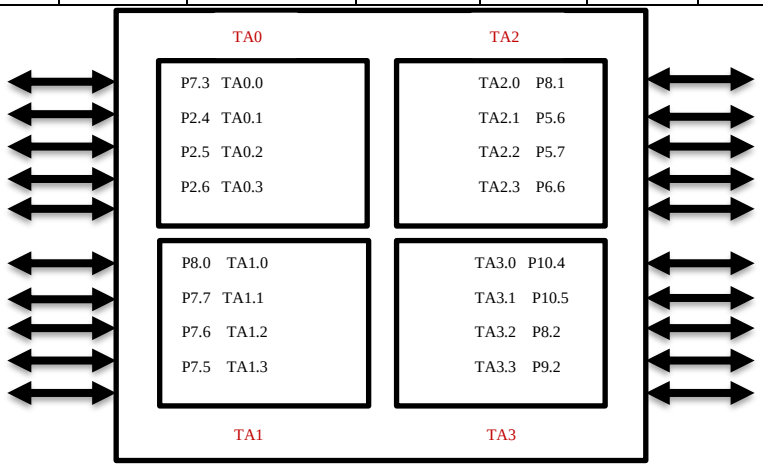
TIMER32_y->LOAD	Load register
TIMER32_y->INTCLR	Clear flag
TIMER32_y->VALUE	Check the counter value

TIMER A REGISTERS

Bits	9-8	7-6	5-4	2	1	0
TIMER_A**y**->CTL	**TASSEL**	**ID**	**MC**	**TACLR**	**TAIE**	**TAIFG**
TIMER_A_CTL_**XX**_x	**0-TACLK,** **1-ACLK** **2-SMCLK,** **3-INCLK**	**0-/1, 1=/2,** **2=/4, 3=/8**	**0-stop,** **1-up** **2- continuous,** **3- up/down**	**1-clear counter**	**1-enable timer interrupt**	**1-timer overflow flag set**

Bits	15-14	13-12	11	10	8	7-5	4	3	2	1	0
TIMER_Ay->CCTL	**CM**	**CCIS**	**SCS**	**SCCI**	**CAP**	**OUTMOD**	**CCIE**	**CCI**	**OUT**	**COV**	**CCIFG**
TIMER_A_CCTLN_**XX_x**	**0-no edge, 1- rising, 2-falling, 3- both**	**0-CCInA, 1-CCInB, 2-GND, 3-VCC**	**1-Synchronize timer clock**	**1-Observe synchronized input**	**1-capture 0-compare**	**0-output, 1-set, 2-toggle/reset, 3- set/reset, 4-toggle, 5-reset, 6- toggle/set, 7-reset/set**	**1-enable interrupt**	**Capture input value**	**Bit value**		**Set for capture in capture mode Set if compare is true in compare**

Bits	3-0
TIMER_Ay->EX0	**TAIDEX**
TIMER_A_EX0_**XX**_x	**0-/1, 1=/2, 2=/3, 3=/4, 4=/5, 5=/6, 6=/7, 7=/8.**



8	Timer_A0	TA0_0_IRQn	TA0_0_IRQHandler	TA0CCR0-CCIFG
9	Timer_A0	TA0_N_IRQn	TA0_N_IRQHandler	TA0CCR1-6, TAIFG
10	Timer_A1	TA1_0_IRQn	TA1_0_IRQHandler	TA1CCR0-CCIFG
11	Timer_A1	TA1_N_IRQn	TA1_N_IRQHandler	TA1CCR1-6, TAIFG
12	Timer_A2	TA2_0_IRQn	TA2_0_IRQHandler	TA2CCR0-CCIFG
13	Timer_A2	TA2_N_IRQn	TA2_N_IRQHandler	TA2CCR1-6, TAIFG
14	Timer_A3	TA3_0_IRQn	TA3_0_IRQHandler	TA3CCR0-CCIFG
15	Timer_A3	TA3_N_IRQn	TA3_N_IRQHandler	TA3CCR1-6, TAIFG

I2C communication

Bits	15	14	13	11	10-9
EUSCI_Bx->CTLW0	UC**A10**	UC**SLA10**	UC**MM**	UC**MST**	UC**MODEx**
EUSCI_B_CTLW0_XX_x	1- select the own address length of the device 0-seven bit address	1-set the slave address	1-choose the master device number(>1) 0- one master in the system	1- used as master, 0- slave	3-I2C
8	7-6	4	3	2	1
UC**SYNC**	UC**SSELx**	UC**TR**	UC**TXNACK**	UC**TXSTP**	UC**TXSTT**
0-asynchronus 1-synchronous	0-UCLKI, 1-ACLK, 2-SMCLK	1-transmitter 0-receiver	1-NACK bit is generated 0-ACK bit occurs	Stop bit generated by master	Start bit generated by master
0 UCSWRST(XX_x=SWRST) 1- reset the module 0-operation mode					

Bits	6	5	3	2	1	0
EUSCIBx->IE	UC**BCNTIE**	UC**NACKIE**	UC**STPIE**	UC**STTIE**	UC**TXIE0**	UC**RXIE0**
EUSCI_B_IE_XX	1-enable byte counter interrupt	1-enable not acknowledge interrupt)	1-enable stop	1-enable start conditions	1-anable transmit interrupt	1-enable receive interrupt

Bits	6	5	3	2	1	0
EUSCIBx->IFG	UC**BCNTIFG**	UC**NACKIFG**	UC**STPIFG**	UC**STTIFG**	UC**TXIFG0**	UC**RXIFG0**
EUSCI_B_IFG_XX	1-set for byte counter interrupt	1-set not acknowledge interrupt)	1-set for stop conditions	1- set start conditions	1-set transmit interrupt	1-set receive interrupt

20	eUSCI_B0	ISER[0]-ICER[0]	0x00100000	EUSCIB0_IRQn	EUSCIB0_IRQHandler
21	eUSCI_B1	ISER[0]-ICER[0]	0x00200000	EUSCIB1_IRQn	EUSCIB1_IRQHandler
22	eUSCI_B2	ISER[0]-ICER[0]	0x00400000	EUSCIB2_IRQn	EUSCIB2_IRQHandler
23	eUSCI_B3	ISER[0]-ICER[0]	0x00800000	EUSCIB3_IRQn	EUSCIB3_IRQHandler

PIN	**PxSEL1=0, PxSEL0=1**
P6.5	UCB1SCL
P6.4	UCB1SDA
P1.6	UCB0SDA
P1.7	UCB0SCL
P6.6	UCB3SDA
P6.7	UCB3SCL
P3.6	UCB2SDA
P3.7	UCB2SCL