**Task 5**

6 *×* 2 = 10 *marks*

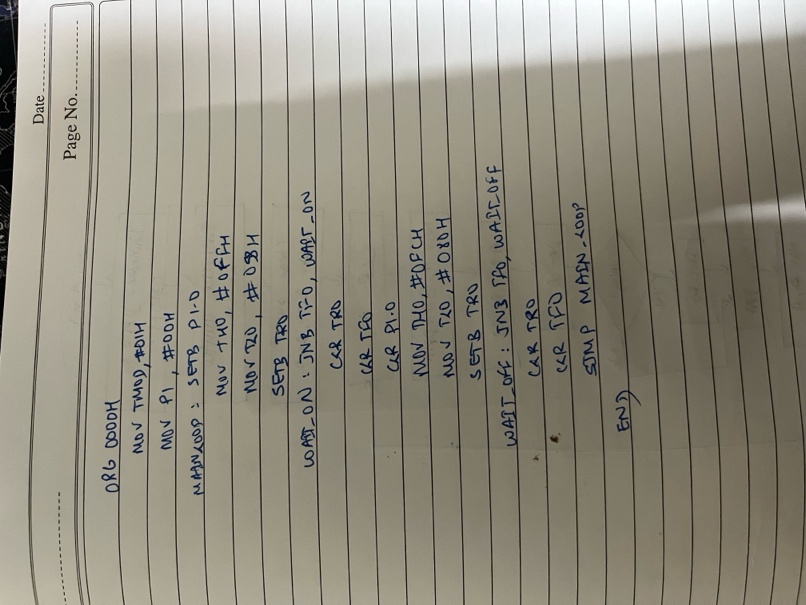
**Name and Registration number: 22BCE3799 Apurba Koirala**



**The task files should have handwritten flow chart/Algorithm, and written Program, Snapshot of typed program and Snapshot of output.**

1. Write an assembly language program to generate a square wave of 1KHz with 20% duty cycle using timer programming.

Written code and flowchart

A diagram on a white paper

AI-generated content may be incorrect.

Code and Output:

ORG 0000H

MOV TMOD, #01H

MOV P1, #00H

MAIN\_LOOP:

SETB P1.0

MOV TH0, #0FFH

MOV TL0, #058H

SETB TR0

WAIT\_ON:

JNB TF0, WAIT\_ON

CLR TR0

CLR TF0

CLR P1.0

MOV TH0, #0FCH

MOV TL0, #080H

SETB TR0

WAIT\_OFF:

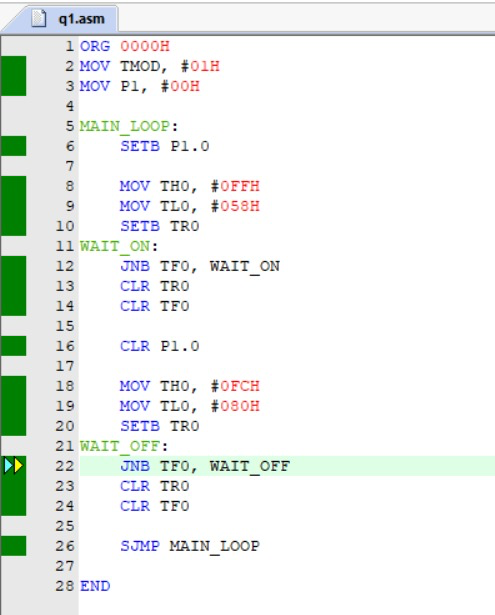
JNB TF0, WAIT\_OFF

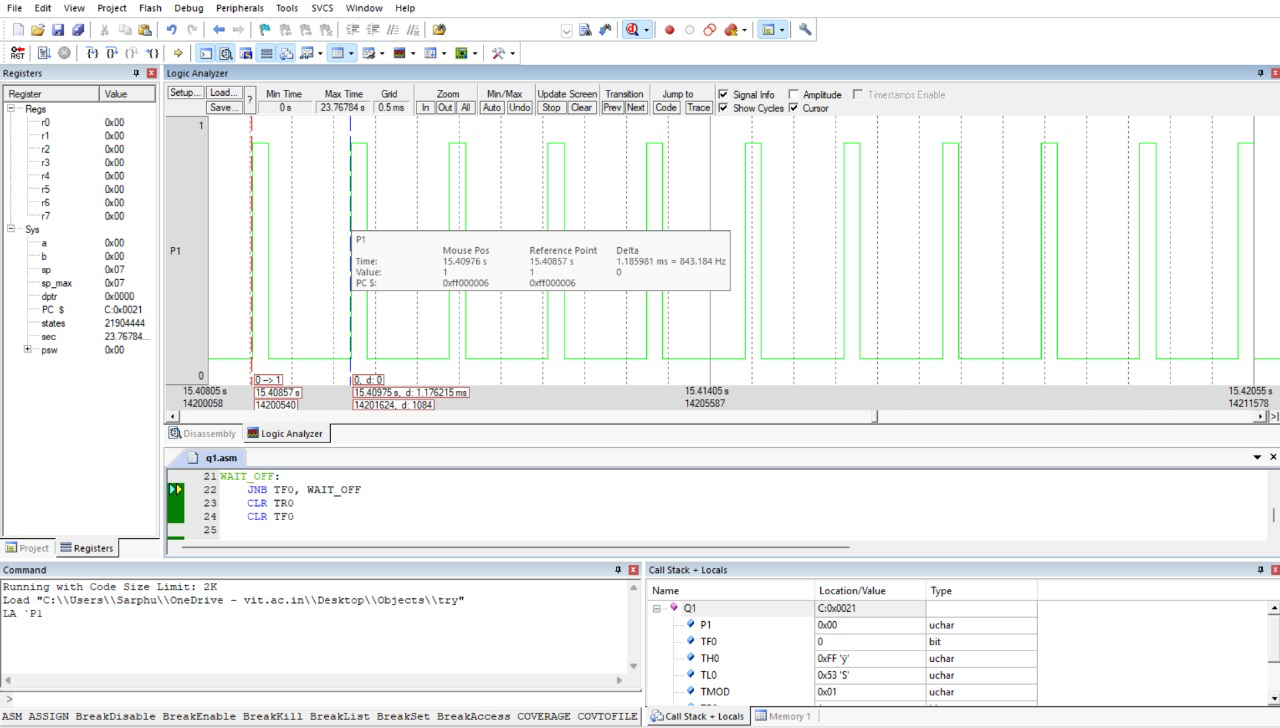
CLR TR0

CLR TF0

SJMP MAIN\_LOOP

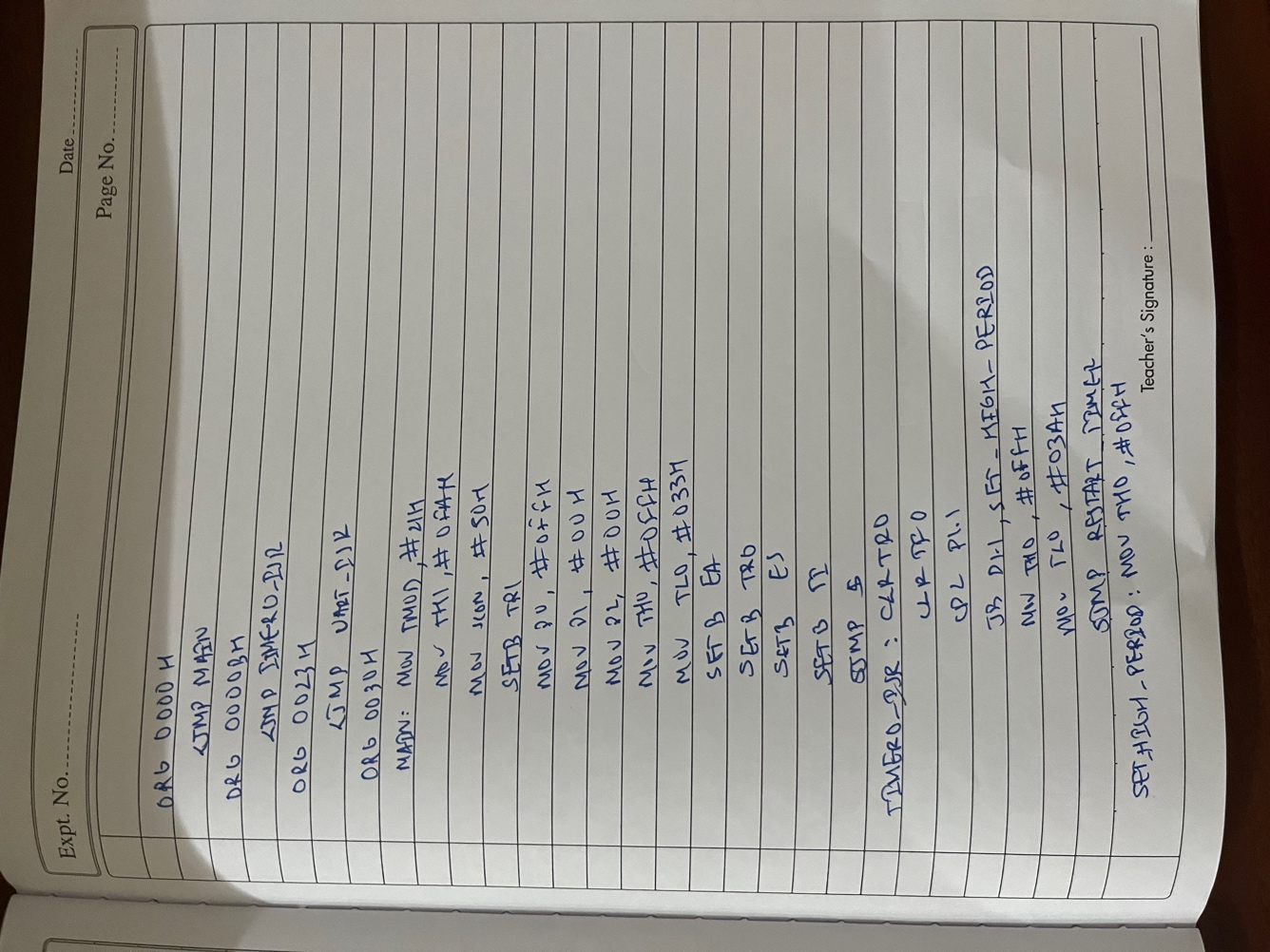
END

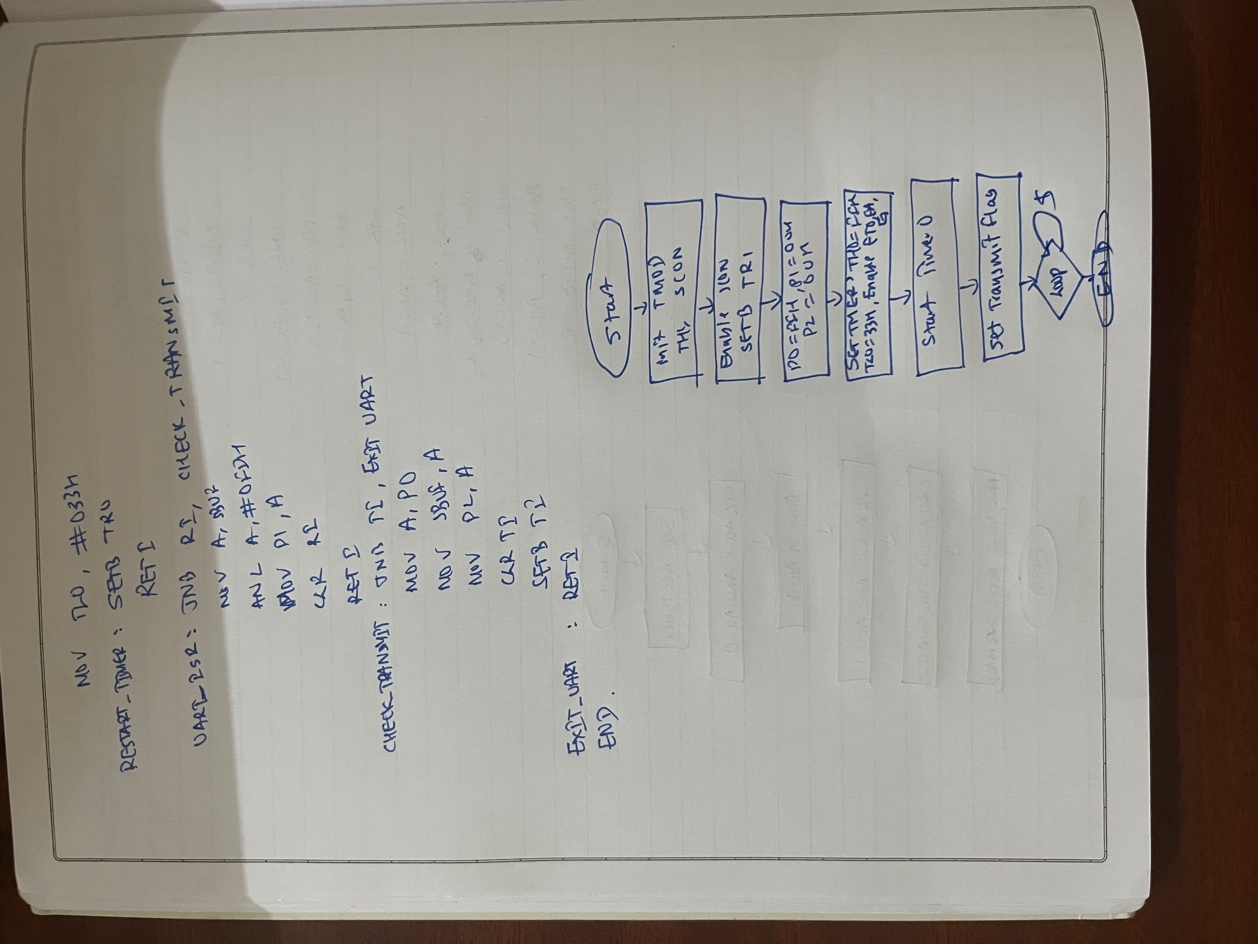




1. Write an assembly language program using interrupts to do the following operations simultaneously: (a) Receive the data serially and send it to P1 (b) Have port P0 read and transmitted serially and a copy given to P2 (c) Make Timer 0 to generate a square wave of 3KHz frequency on P1.1. with 66.67% duty cycle. Assume that XTAL = 11.0592Mhz. Set the baud rate at 4800.

Handwritten code and flowchart





Code and Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

ORG 0000H

LJMP MAIN

ORG 000BH

LJMP TIMER0\_ISR

ORG 0023H

LJMP UART\_ISR

ORG 0030H

MAIN:

MOV TMOD, #21H

MOV TH1, #0FAH

MOV SCON, #50H

SETB TR1

MOV P0, #0FFH

MOV P1, #00H

MOV P2, #00H

MOV TH0, #0FFH

MOV TL0, #033H

SETB ET0

SETB EA

SETB TR0

SETB ES

SETB TI

SJMP $

TIMER0\_ISR:

CLR TR0

CLR TF0

CPL P1.1

JB P1.1, SET\_HIGH\_PERIOD

MOV TH0, #0FFH

MOV TL0, #09AH

SJMP RESTART\_TIMER

SET\_HIGH\_PERIOD:

MOV TH0, #0FFH

MOV TL0, #033H

RESTART\_TIMER:

SETB TR0RETI

UART\_ISR:

JNB RI, CHECK\_TRANSMIT

MOV A, SBUF

ANL A, #0FDH

MOV P1, A

CLR RI

RETI

CHECK\_TRANSMIT:

JNB TI, EXIT\_UART

MOV A, P0

MOV SBUF, A

MOV P2, A

CLR TI

SETB TI

EXIT\_UART:

RETI

END