**MICRONCONTROLLERS AND ITS APPLICATIONS LAB**

## **NAME: Pratyush Raj**

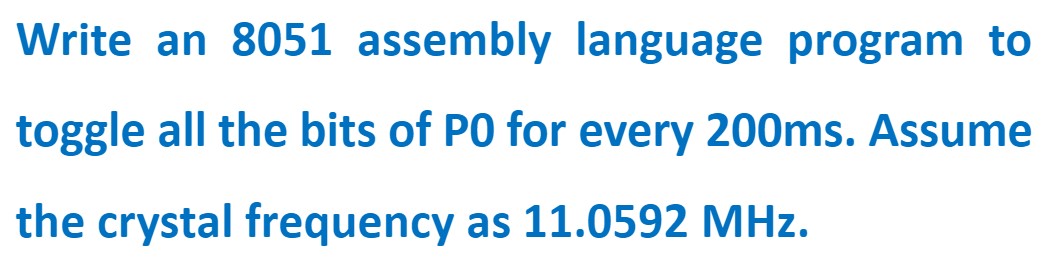
## **REG.N0: 20BEC1273**

## **DATE: 07-03-2022**

EXP 5: ASSEMBLY PROGRAM WITH I/O PORTS OF 8051

**LAB** **TASK**-**1**

**AIM:**



**SOFTWARE USED:** Keil IDE

**PROGRAM:**

ORG 0000H

MOV P0,#00H

BACK: MOV A,#55H

MOV P0, A

ACALL DELAY

MOV A,#0AAH

MOV P0,A

ACALL DELAY

SJMP BACK

DELAY: MOV R2, #200

AGAIN: MOV R3, #250

HERE: NOP

NOP

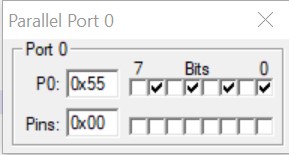
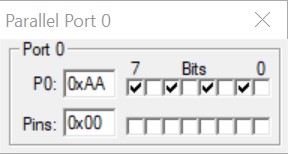
DJNZ R3, HERE

DJNZ R2, AGAIN

RET

END

**OUTPUT**:



The results keep changing between memory locations 0x55 and 0xAA with a time interval of 200ms.

**RESULT**: We have successfully written a program to toggle the bits of P0 for every 200ms.

**LAB TASK 2**

**AIM**:



ORG 000H

# BACK: SETB P1.3

# ACALL DELAY

ACALL DELAY

CLR P1.3

ACALL DELAY

ACALL DELAY

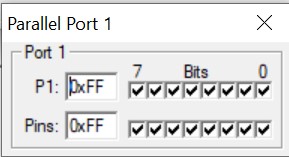
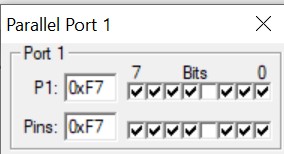
SJMP BACK

DELAY: MOV R3, #200

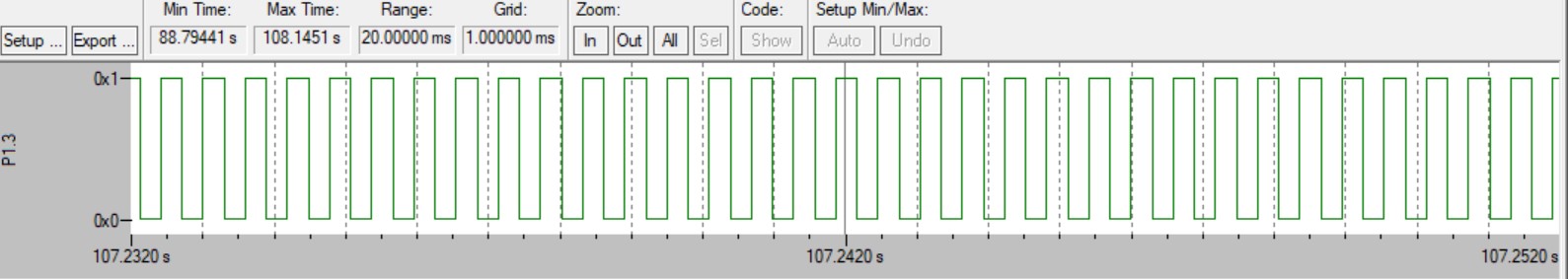
HERE: DJNZ R3, HERE

RET END

**OUTPUT**:



The results keep changing between memory locations 0xF7 and 0xFF.



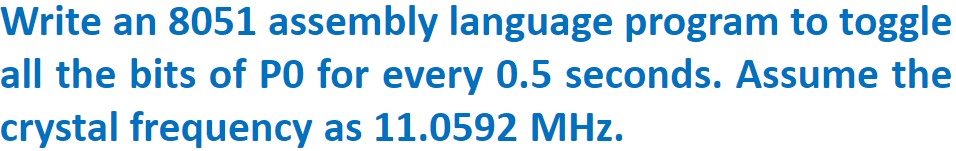
The above picture gives us the logic analysis - graph for the problem.

**RESULT**: We have successfully written a program to create a square wave of 25% duty cycle on bit 3 of port1.

**CHALLENGING TASKS**

**TASK 1**

**AIM**:



ORG 0000H

MOV P0, #00H

BACK: MOV A,#55H

MOV P0, A

ACALL DELAY

MOV A,#0AAH

MOV P0, A

ACALL DELAY

SJMP BACK

DELAY: MOV R2, #500

AGAIN: MOV R3, #550

HERE: NOP

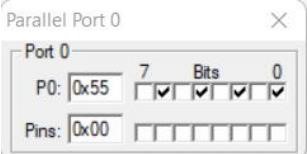
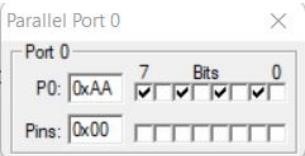
NOP

DJNZ R3, HERE

DJNZ R2, AGAIN

RET END

**OUTPUT**:



The results keep changing between memory locations 0x55 and 0xAA with a time interval of 0.5 seconds.

**RESULT**: We have successfully written a program to toggle all the bits of P0 for every 0.5 seconds.

**TASK 2 AIM:**



ORG 0000H

MOV P1,#00H

BACK: SETB P1.3

ACALL DELAY

ACALL DELAY

ACALL DELAY

CLR P1.3

ACALL DELAY

SJMP BACK

DELAY: MOV R2,#04H

HERE3:MOV R1,#0FFH

HERE2:MOV R0,#0FFH

HERE1: DJNZ R0,HERE1

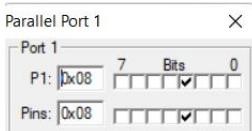
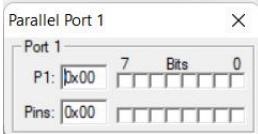
DJNZ R1,HERE2

DJNZ R2,HERE3

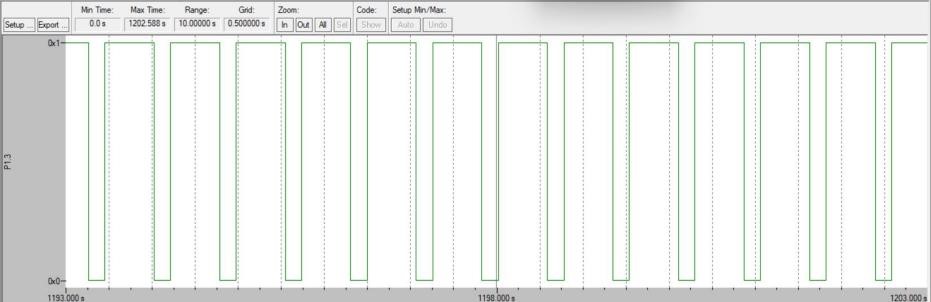
RET

END

**OUTPUT**:



The results keep changing between memory locations 0x00 and 0x08.



The above picture gives us the logic analysis - graph for the problem.

**RESULT**: We have successfully written a program to create a square wave of 75% duty cycle on bit3 on port1.

--------------------------------------------------------------------------------------------------------