

# Microprocessors and Microcontrollers

## Assignment

Samriddha Chattopadhyay

200020040

1. Write a program to find the average of 5 numbers stored in memory locations starting from 60H

```
# ORG 0200H
    MVI C,05      ;for taking 5 numbers
    MVI B,05      ;Counter
    MVI D,00      ;storing quotient
    LXI H,0060    ;taking value from 0060H
    MOV E,M       ;reading from memory

LOOP1:    MOV A,E  ;move data to accumulator for further addition
    INX H      ;increment the pointer to load data
    ADD M      ;add memory to accumulator
    MOV E,A    ;move the sum to E to store it
    DCR B      ;decrement the counter
    MOV A,B    ;move the counter to accumulator to determine if the loop should keep
running
    JNZ LOOP1  ;if the counter is not zero, keep the loop running
    MOV A,E    ;move the total sum to accumulator for further operations
    JZ LOOP2   ;jump to Loop 2 if counter goes to zero

LOOP2:    CMP C  ;compare with C to find the average
    JC LOOP3   ;store the answer if the value goes to zero
    SUB C      ;subtract to get the average
    INR D      ;increase the count
    JMP LOOP2  ;loop back

LOOP3:    INX H
    MOV A,D
    STA 2500
    HLT
# ORG 0060H
# DB 10H,15H,20H,25H,06H
```

2. Write an assembly language program to perform division of two numbers in R0 and R1. The quotient should be stored in A and the remainder in B. DO NOT USE the DIV AB instruction! Write comments for each line of the code.

org 0000h

```
MOV R0,#0FH ;DIVIDEND
MOV R1,#03H ;DIVISOR
MOV R7,#00H ;QUOTIENT
MOV R4,#00H ;REMAINDER
MOV A,R0 ;MOVE DIVIDEND IN A
```

LOOP:

```
SUBB A,R1 ;SUBTRACT THE DIVISOR FROM THE DIVIDEND
MOV R4,A ;STORE THE REMAINDER
JC BREAK ;IF THE DIVIDEND BECOMES 0, BREAK THE LOOP AND SAVE THE
RESULTS
INC R7 ;STORE THE QUOTIENT
SJMP LOOP ;LOOP BACK
```

BREAK:

```
ADD A,R1 ;PUT THE QUOTIENT IN A
MOV B,A ;MOVE A INTO B TO SAVE IT
MOV A,R7 ;PUT THE REMAINDER IN B
```

END

3. Write a program to generate a 2.5 kHz waveform on P1.0 with a 20% duty cycle. Write comments for each line of the code.

MAIN:

```
CLR P1.0    ;PORT1.0 IS CLEARED SO IS 0
ACALL DELAY2 ;GO TO DELAY2 FOR CREATE DELAY FOR SIGNAL
SETB P1.0   ;PORT1.0 BECOMES 1
MOV TCON,#00H
ACALL DELAY1 ;GO TO DELAY1 FOR CREATE DELAY FOR SIGNAL 1
ACALL ENDCODE
```

WAIT1:

```
JNB TF0, WAIT1
CLR TR0
CLR TF0
RET
```

WAIT2:

```
JNB TF0, WAIT2
CLR TR0
CLR TF0
RET
```

DELAY1:

```
MOV TMOD,#01H ; SELECT MODE OF OPERATION
               ;COUNT CALCULATED=MAX COUNT(65535) - REQUIRED COUNT(80) + 1
MOV TL0,#0B0H ; LOAD LOWER NIBBLE /8BITS IN TL0
MOV TH0,#0FFH ;LOAD UPPER NIBBLE /8BITS IN TH0
MOV TCON,#10H
```

DELAY2:

```
MOV TMOD,#01H ;SELECT MODE OF OPERATION
               ;COUNT CALCULATED=MAX COUNT(65535) - REQUIRED COUNT(320) +
1
MOV TL0,#0C0H ;LOAD LOWER NIBBLE /8BITS IN TLO
MOV TH0,#0FEH
MOV TCON,#10H
```

ENDCODE:

```
END
```