

Microprocessors and Microcontrollers Lab

Digital Assignment 1

ADARSH ARUNKUMAR SHIRAWALMATH

22BKT0058

Aim:

- i.) To create an Assembly Language Program to find the sum and mean of five numbers, using the 8051 microcontroller**
- ii.) To create an Assembly Language Program to find the variance of marks of ten students, using the 8051 microcontroller**

Procedure:

- i.) Start up the Keil μ Vision Software.
- ii.) Create new μ Vision project at required directory.
- iii.) Set the device as 8051 microcontroller (AT89C51).
- iv.) Create new item at Source Group 1 in Target 1.
- v.) Set the file type as ASM file.
- vi.) Continue writing the code for the ALP.
- vii.) Translate and build the file.
- viii.) Start debug session, and run code line by line to get output
- ix.) Check output at the memory location set, in memory 1.

Algorithm:

- a.) To find the sum and mean of five numbers:-

Set 5 numbers at 5 adjacent memory locations
Set value of Accumulator to 0, and Register 0 to location of first number
Set B to 5.
Add value at location stored in Register 0 to Accumulator.
Increment value of Register 0.
Repeat till all values are added to Accumulator.
Sum of 5 numbers stored at Accumulator
Divide value at Accumulator by value at B.
Output at Accumulator.

b.) To find variance of marks of ten students:-

Store values at 10 adjacent locations
Set R0 to first location
Set A to 0
Set B to 10
Loop 10 times(
Add value @R0 to A
Increment R0
)
Divide A by B to get mean
Store value of A at R3
Set value at R3 to B
Set value of R1 to memory location
Loop 10 times(
Set value @R0 to A
Subtract A by B

```
Store value at A, at memory location at R1
Increment R1
)
Reset R1 to original memory location
Loop 10 times(
Set A and B to value @R1
Multiply A and B
Set value @R1 to value at A
Increment R1
)
Set A to 0
Set B to 10
Loop 10 times(
Add value @R1 to A
Increment R1
)
Divide A by B
Output at Accumulator
```

Code:

a.) To find sum and mean of 5 numbers:-

```
1  MOV 30H, #05H
2  MOV 31H, #0aH
3  MOV 32H, #01H
4  MOV 33H, #09H
5  MOV 34H, #0fH
6  MOV R0, #30H
7  MOV A, #00H
8  MOV B, #5H
9
10 ADD A, @R0
11 INC R0
12 ADD A, @R0
13 INC R0
14 ADD A, @R0
15 INC R0
16 ADD A, @R0
17 INC R0
18 ADD A, @R0
19 INC R0
20
21 DIV AB
22 END
```

b.) To find variance of 10 numbers

```
1  MOV 30H, #05H
2  MOV 31H, #0aH
3  MOV 32H, #01H
4  MOV 33H, #09H
5  MOV 34H, #0fH
6  MOV 35H, #07H
7  MOV 36H, #08H
8  MOV 37H, #0cH
9  MOV 38H, #01H
10 MOV 39H, #06H
11
12 MOV R0, #30H
13 MOV A, #00H
14 MOV B, #0aH
15
16 MOV R4, #0aH
17 LOOP: MOV A, @R0
18 INC R0
19 DJNZ R4, LOOP
20 DIV AB
21 MOV R3, A
22
23 MOV R4, #0aH
24 MOV R0, #30H
25 MOV R1, #40H
26 MOV A, #00H
27 LOOP1: MOV A, @R0
28 MOV B, R3
29 SUBB A, B
```

```

30 MOV @R1, A
31 INC R0
32 INC R1
33 DJNZ R4, LOOP1
34
35 MOV R4, #0aH
36 MOV R0, #30H
37 MOV R1, #40H
38 LOOP2: MOV A, @R1
39 MOV B, @R1
40 MUL AB
41 MOV @R1, A
42 INC R1
43 DJNZ R4, LOOP2
44
45 MOV A, #00H
46 MOV B, #0aH
47 MOV R4, #0aH
48 MOV R1, #40H
49 LOOP3: ADD A, @R1
50 INC R1
51 DJNZ R4, LOOP3
52
53 DIV AB
54
55 END

```

Output:

a.) Sum and Mean of 5 numbers

Registers

Register	Value
r0	0x35
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x08
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0021
states	28
sec	0.000014...
psw	0x41

Disassembly

```

C:0x0021 00 NOP
C:0x0022 00 NOP
C:0x0023 00 NOP
C:0x0024 00 NOP

```

NewProj.a51

```

1 MOV 30H, #05H
2 MOV 31H, #0aH
3 MOV 32H, #01H
4 MOV 33H, #09H
5 MOV 34H, #0fH
6 MOV R0, #30H
7 MOV A, #00H
8 MOV B, #5H
9
10 ADD A, @R0
11 INC R0
12 ADD A, @R0
13 INC R0
14 ADD A, @R0
15 INC R0
16 ADD A, @R0
17 INC R0
18 ADD A, @R0
19 INC R0
20
21 DIV AB
22 END

```

b.) Variance of 10 numbers

Registers

Register	Value
r0	0x30
r1	0x4a
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x15
b	0x04
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x005B
states	326
sec	0.00016300
psw	0x01

Disassembly

```

C:0x005B 00 NOP
C:0x005C 00 NOP
C:0x005D 00 NOP
C:0x005E 00 NOP

```

NewProj.a51

```

34
35 MOV R4, #0aH
36 MOV R0, #30H
37 MOV R1, #40H
38 LOOP2: MOV A, @R1
39 MOV B, @R1
40 MUL AB
41 MOV @R1, A
42 INC R1
43 DJNZ R4, LOOP2
44
45 MOV A, #00H
46 MOV B, #0aH
47 MOV R4, #0aH
48 MOV R1, #40H
49 LOOP3: ADD A, @R1
50 INC R1
51 DJNZ R4, LOOP3
52
53 DIV AB
54
55 END

```