VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering ,VIT, Vellore

Reg.No	19BEC0358		
Student Name	ARPIT PATAWAT		
Course Code	ECE3003	**Slot & Semester**	L35+L36 FALL -- 2021-22
Course Name	Microcontroller and its applications		
Program Title	Lab Assignment 1(ADD BCD STACK PROGRAMS)		
Date of Exp.	16-08-21	**Date of Submission**	30-08-21
Faculty	A.Karthikeyan		

Question

1. Write an 8051 ALP to add the following data and then use the simulator to examine the CY flag.

a) INPUT DATA YOUR REG NO: 19BEC0358 Example 01 9B EC 03 58.

b) Five max 8BIT NUMBERS (FF, FF, FF, FF, FF)

2. Write an 8051 ALP assemble a program to load values into each of registers R0 – R5 and then push each of these registers onto the stack. Single-step the program, and examine the stack and the SP register after the execution of each instruction.

3. Write an 8051 assemble language program to:

(a) Set SP = 0D,

(b) Load a different value in each of RAM locations 0D, 0C, 0B, 0A, 09, and 08,

(c) POP each stack location into registers R0 - R4. Use the simulator to single-step and examine the registers, the stack, and the stack pointer.

4. Write an 8051 ALP assemble a program to load values into each of registers R0 - R4 and then push each of these registers onto the stack and pop them back. Single-step the program, and examine the stack and the SP register after the execution of each instruction.

5.Write an 8051 ALP to add 10 bytes of BCD data and store the result in R2 and R3.
The bytes are stored in ROM space starting at 300H.

The data would look as follows:
MYDATA: DB ;

Input: i) (10 DATA) : 22H,43H,23H,34H,31H,77H,91H,33H,43H,07H

ii) (10 DATA) : YOUR REG NO: _____

Example 01 7B EC 02 75 00 00 00 00 00

Notice that you must first bring the data from ROM space into the CPU's RAM,
then add them together. Use a simulator to single-step the program and examine the data.

TASK 1.a) -

Aim: To write an 8051 ALP to perform addition of 5 8BIT NUMBERS (Reg. No.) using keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.Load first 2 value out of 5 to operands A and B.

2.Add both the operands and if carry flag is 1 then increment R0.

3.load the value of A to B and then load the third value to A.

4.Again add both the values and increment R0 if carry flag is 1.

5.Repeat the process until we add all the 5 values.

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG 0000h						Assembler directive	NONE
	MOV	A, #001H	Immediate	1	2	Data transfer	Load FF to A	PARITY
	MOV	B, #09BH	Immediate	1	2	Data Transfer	Load FF to B	NONE
	ADD	A, B	Register	1	1	Arithmetic	Add content of A and B and store it in A	PARITY,
	JNC	L1	Indexed	2	2	Boolean	Jump to label L1 if there is no carry	NONE
	INC	R0	Register	1	1	Arithmet	Register	NONE

						ic	R0 is increment ed by 1	
L1	MOV	B, A	Register	1	1	Data transfer	Load the content of A into B	NONE
	MOV	A, #0ECH	Immediate	1	2	Data transfer	Load FF to A	PARITY
	ADD	A, B	Register	1	1	Arithmet ic	Add content of A and B and store it in A	PARITY, AUXILA RY CARRY FLAG, CARRY FLAG
	JNC	L2	Indexed	2	2	Boolean	Jump to label L2 if there is no carry	NONE
	INC	R0	Immediate	1	1	Arithmet ic	Register R0 is increment ed by 1	NONE
L2	MOV	B, A	Register	1	1	Data transfer	Load the content of A into B	NONE
	MOV	A, #003H	Immediate	1	2	Data transfer	Load FF to A	NONE
	ADD	A, B	Register	1	1	Arithmet ic	Add content of A and B and store it in A	AUXILA RY CARRY FLAG, CARRY FLAG
	JNC	L3	Indexed	2	2	Boolean	Jump to label L3 if there is no carry	NONE

	INC	R0	Immediate	1	1	Arithmetic	Register R0 is incremented by 1	NONE
L3	MOV	B, A	Register	1	1	Data transfer	Load the content of A into B	NONE
	MOV	A, #058H	Immediate	1	2	Data transfer	Load FF to A	PARITY
	ADD	A, B	Register	1	1	Arithmetic	Add content of A and B and store it in A	AUXILARY CARRY FLAG
	JNC	L4	Indexed	2	2	Boolean	Jump to label L4 if there is no carry	NONE
	INC	R0	Immediate	1	1	Arithmetic	Register R0 is incremented by 1	NONE
L4							Label L4	
	END						End of programme	

Output: Registers containing the Result: R0=01H, A=E3H

Manual Calculation: hexa decimal to binary conversion ->

01 = 1, 9B = 155, EC =236, 03 = 3, 58 = 88

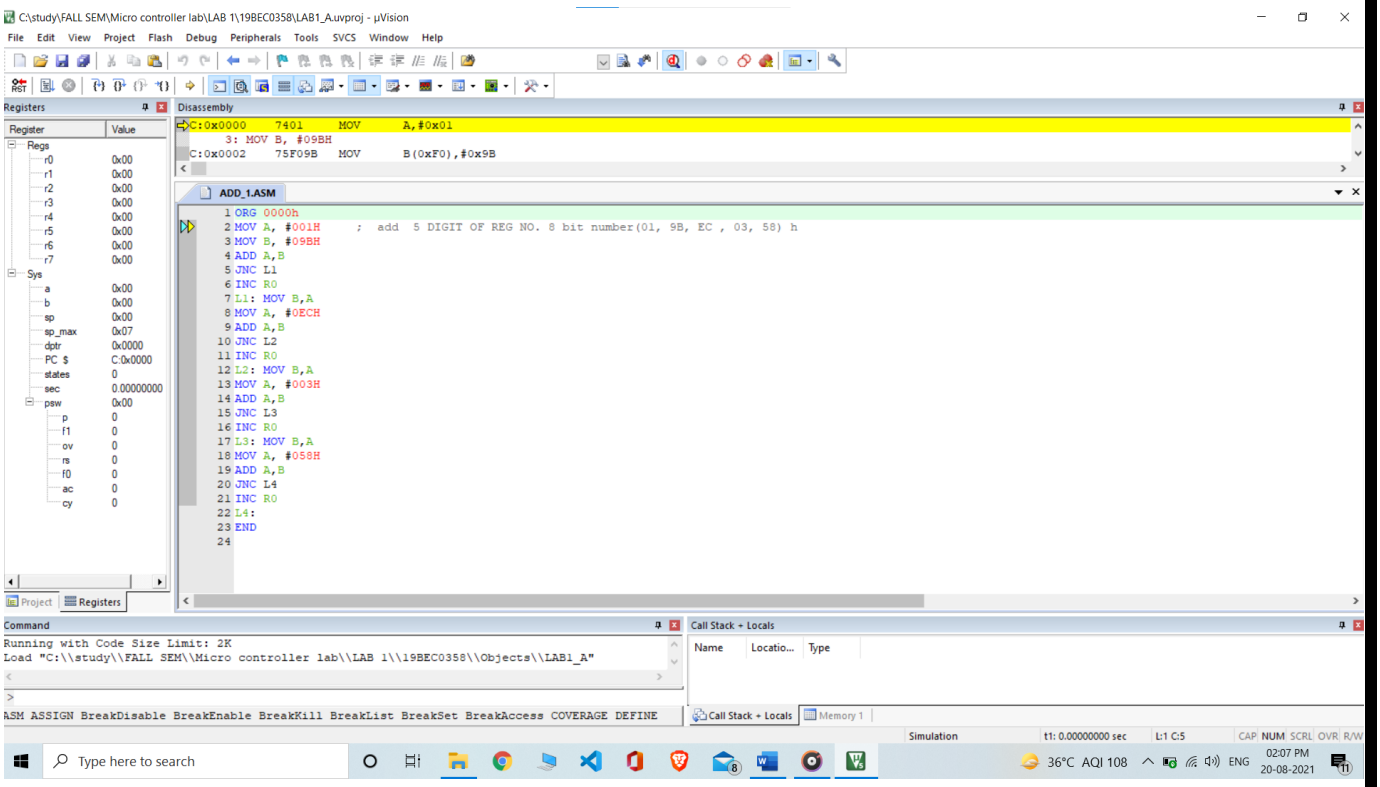
1+ 155 + 236 + 3 + 88 = 483

483 = 1E3 in hexa decimal

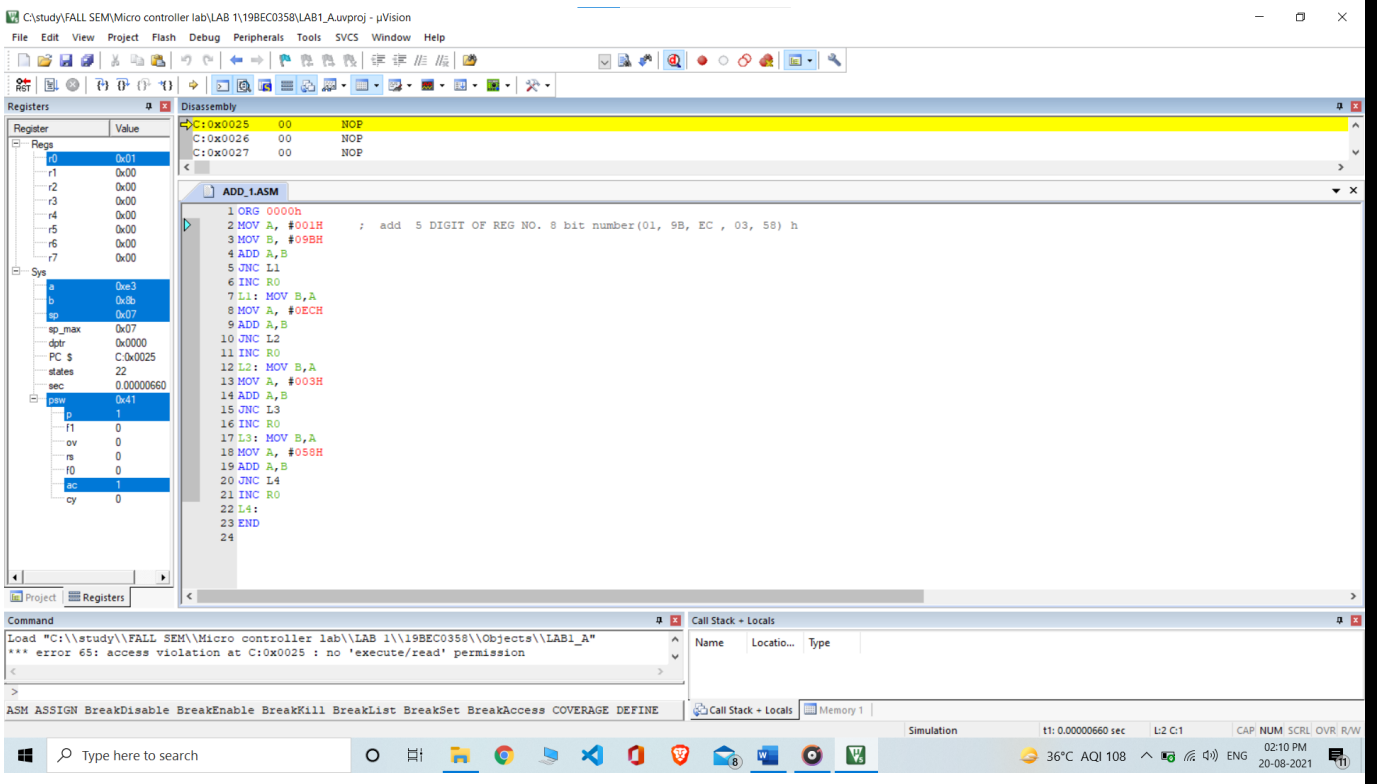
Answer – **1E3**

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – Parity flag is 1 since there are odd numbers of 1 in E3, auxiliary carry flag is Also 1 since in the last addition so there is a carry from D3 to D4 and also there is no carry Flag.

2. ABOUT THE OUTPUT VALUES IN REGISTERS – every time when there is carry, R0 is incremented that's why we have R0 = 1 only.

Result: The 8051 ALP to perform addition of 5 8BIT NUMBERS is executed using Keil software and the results are verified manually.

----------XXXXXXX----------

TASK 1.b) -

Aim: To write an 8051 ALP to perform addition of 5 8BIT NUMBERS using Keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.Load first 2 value out of 5 to operands A and B.

2.Add both the operands and if carry flag is 1 then increment R0.

3.load the value of A to B and then load the third value to A.

4.Again add both the values and increment R0 if carry flag is 1.

5.Repeat the process until we add all the 5 values.

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG 0000h						Assembler directive	NONE
	MOV	A, #0FFH	Immediate	1	2	Data transfer	Load FF to A	NONE
	MOV	B, #0FFH	Immediate	1	2	Data Transfer	Load FF to B	NONE
	ADD	A, B	Register	1	1	Arithmetic	Add content of A and B and store it in A	PARITY, AUXILARY CARRY FLAG, CARRY FLAG
	JNC	L1	Indexed	2	2	Boolean	Jump to label L1 if there is no	NONE

							carry	
	INC	R0	Immediate	1	1	Arithmetic	Register R0 is incremented by 1	NONE
L1	MOV	B, A	Register	1	1	Data transfer	Load the content of A into B	NONE
	MOV	A, #0FFH	Immediate	1	2	Data transfer	Load FF to A	NONE
	ADD	A, B	Register	1	1	Arithmetic	Add content of A and B and store it in A	PARITY
	JNC	L2	Indexed	2	2	Boolean	Jump to label L2 if there is no carry	NONE
	INC	R0	Immediate	1	1	Arithmetic	Register R0 is incremented by 1	NONE
L2	MOV	B, A	Register	1	1	Data transfer	Load the content of A into B	NONE
	MOV	A, #0FFH	Immediate	1	2	Data transfer	Load FF to A	NONE
	ADD	A, B	Register	1	1	Arithmetic	Add content of A and B and store it in A	PARITY
	JNC	L3	Indexed	2	2	Boolean	Jump to label L3 if there is no carry	NONE

	INC	R0	Immediate	1	1	Arithmetic	Register R0 is incremented by 1	NONE
L3	MOV	B, A	Register	1	1	Data transfer	Load the content of A into B	NONE
	MOV	A, #0FFH	Immediate	1	2	Data transfer	Load FF to A	NONE
	ADD	A, B	Register	1	1	Arithmetic	Add content of A and B and store it in A	PARITY
	JNC	L4	Indexed	2	2	Boolean	Jump to label L4 if there is no carry	NONE
	INC	R0	Immediate	1	1	Arithmetic	Register R0 is incremented by 1	NONE
L4							Label L4	
	END						End of programme	

Output: Registers containing the Result: R0=04H, A=FBH

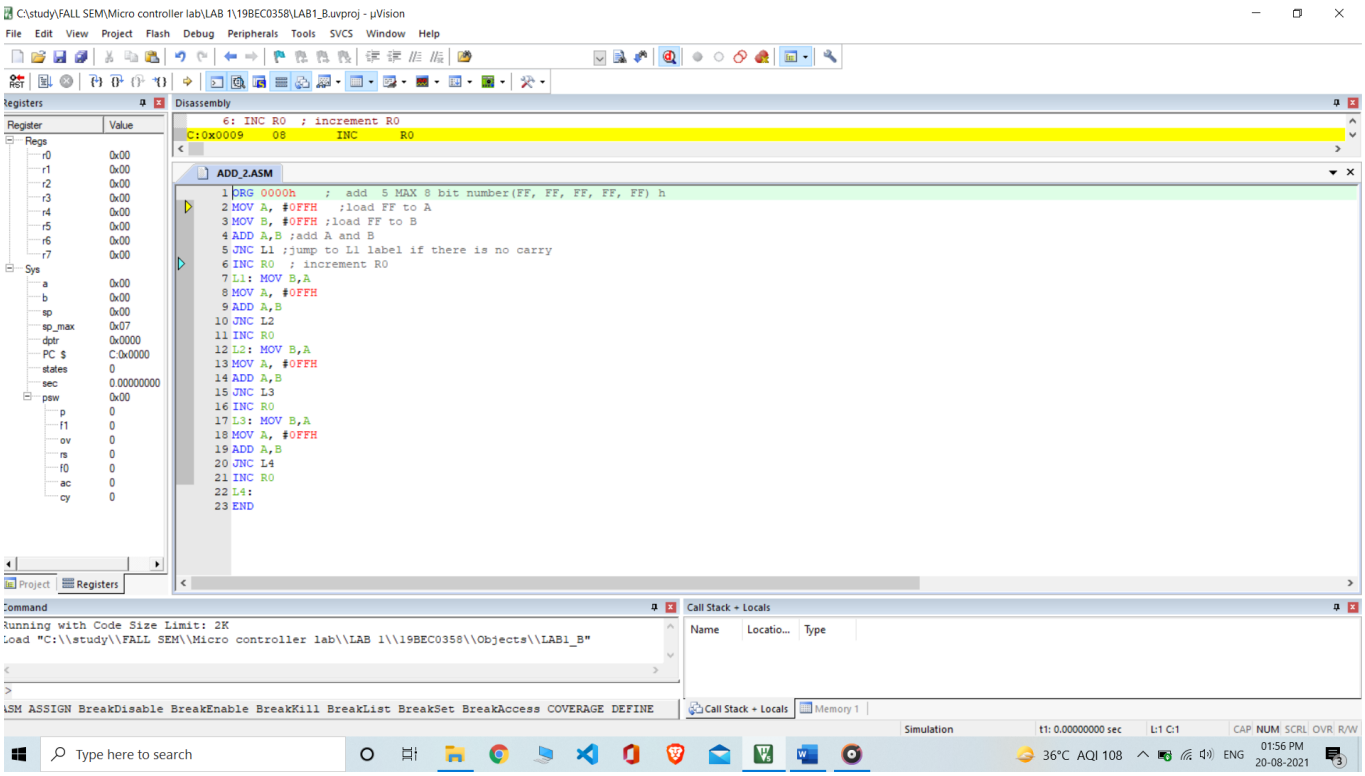
Manual Calculation: FF = 255 (in binary)

FF + FF +FF + FF + FF is -

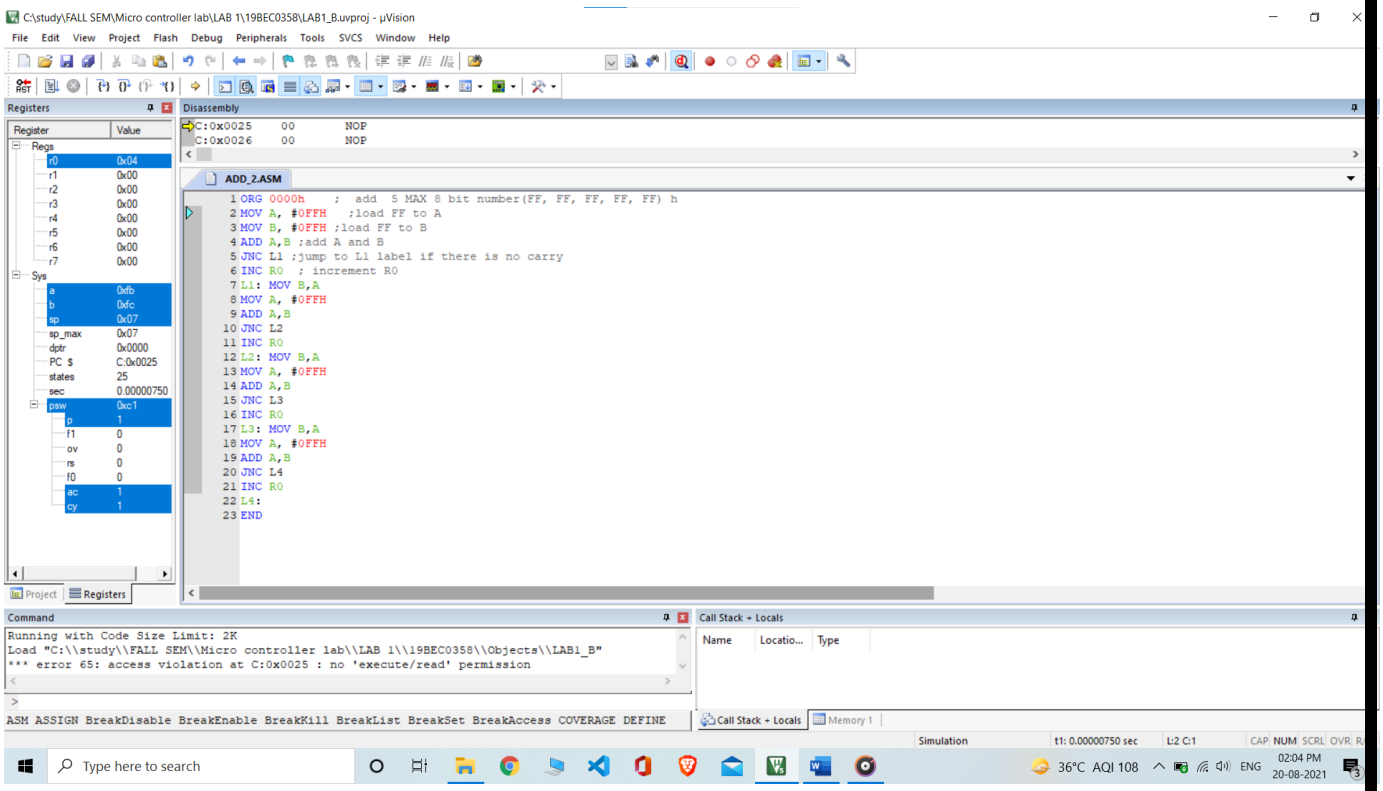
255 + 255+ 255+ 255+ 255 = 1275 in decimal = 4FB in Hexa decimal

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – Parity flag is 1 since there are odd numbers of 1 in FB, auxiliary carry flag is Also 1 since in the last addition for FC+FF, there is a carry from D3 to D4 and also there is carry At D7 so carry flag is also 1.
2. ABOUT THE OUTPUT VALUES IN REGISTERS – every time when there is carry, R0 is incremented that's why we have R0 = 4.

Result: The 8051 ALP to perform addition of 5 8BIT NUMBERS is executed using Keil software and the results are verified manually.

----------XXX----------

TASK 2)

Aim: To write an 8051 ALP to load values from registers and to push them to stack using Keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.load different numbers to registers from R0 - R5

2.push each value into the stack

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000h						
	MOV	R0, #001H	Immediate	1	2	Data transfer	Value is loaded to R0	NONE
	MOV	R1, #09BH	Immediate	1	2	Data transfer	Value is loaded to R1	NONE
	MOV	R2, #0ECH	Immediate	1	2	Data transfer	Value is loaded to R2	NONE
	MOV	R3, #003H	Immediate	1	2	Data transfer	Value is loaded to R3	NONE
	MOV	R4, #058H	Immediate	1	2	Data transfer	Value is loaded to R4	NONE
	MOV	R5, #012H	Immediate	1	2	Data transfer	Value is loaded to R5	NONE
	PUSH	0	Direct	2	2	Data transfer	Value is pushed to	NONE

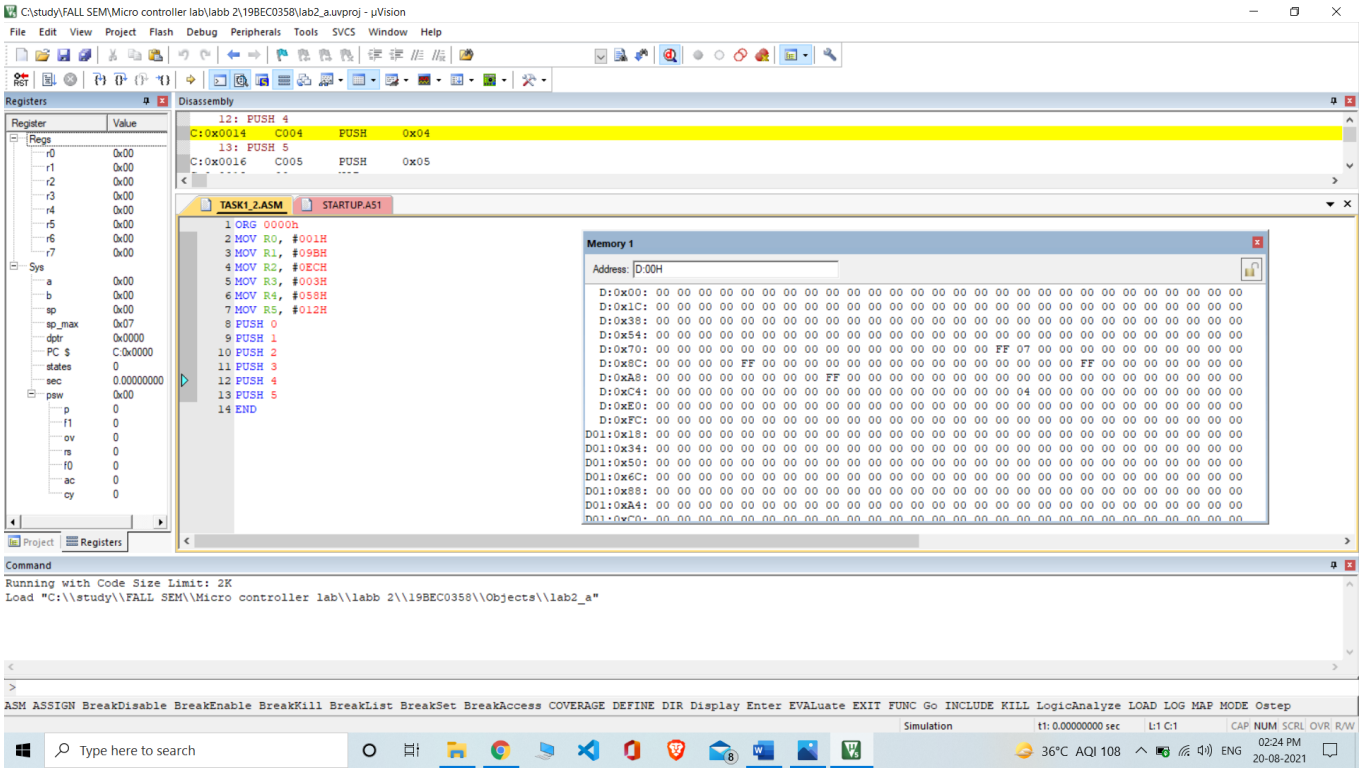
							stack	
	PUSH	1	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	2	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	3	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	4	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	5	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	END						Programme ends here	NONE

Output: Registers containing the Result: R0 =01, R1=9B, R2 =EC, R3=03, R4 = 58, R5 =12

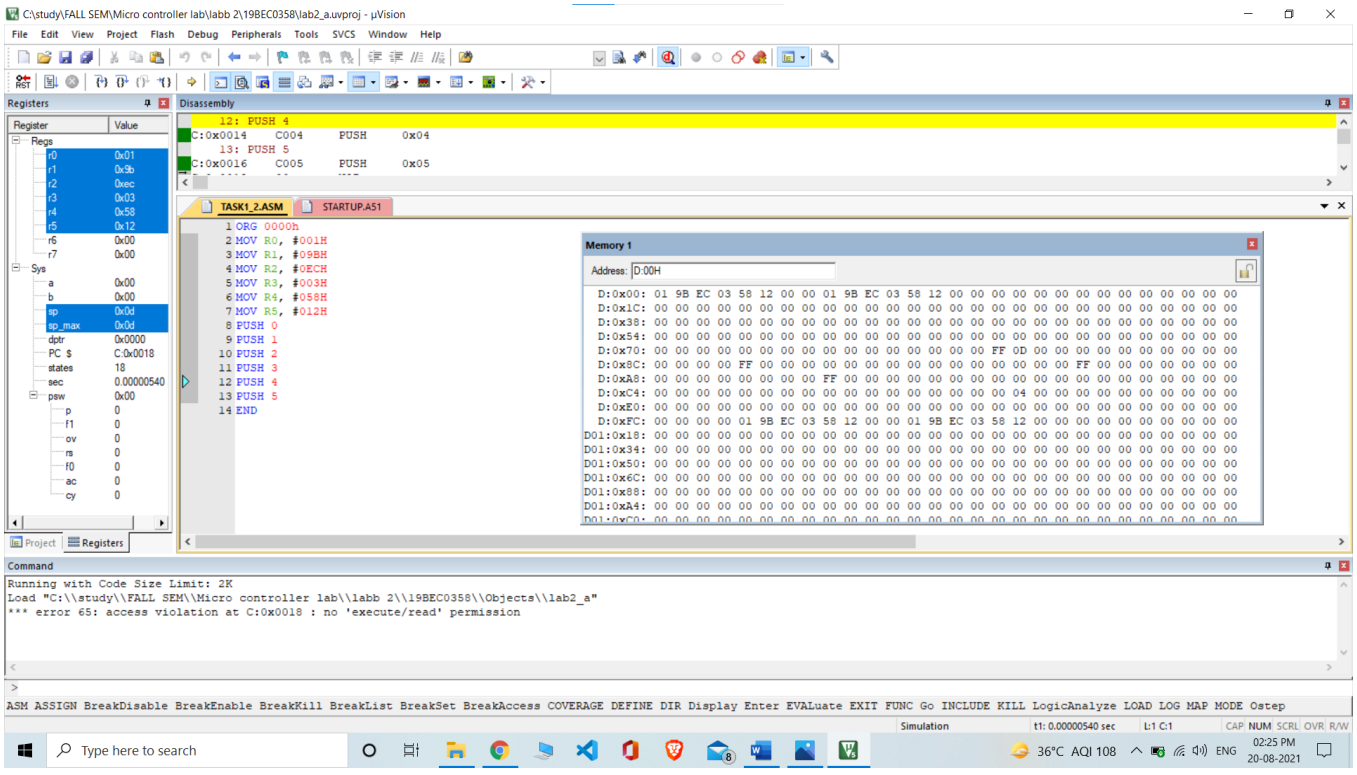
Manual Calculation: NONE

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – All are 0.

2. ABOUT THE OUTPUT VALUES IN REGISTERS – form R0 to R5 all the data is pushed to stack Memory and stack pointer is incremented by 1 whenever data is pushed

Result: The 8051 ALP to PUSH 6 8Bit numbers is executed using Keil software and the results are verified manually.

----------XXX----------

TASK 3)

Aim: To write an 8051 ALP to load values to different RAM locations and POP each value to registers using Keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.Set stack pointer location to 0D.

2.Load a different value at each RAM location.

3.POP each stack location into registers R0 – R4.

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000h						
	MOV	SP, #0DH	Immediate	1	2	Data transfer	Value is loaded to stack pointer	NONE
	MOV	08H, #001H	Immediate	1	2	Data transfer	Value is loaded at the specified location	NONE
	MOV	09H, #09BH	Immediate	1	2	Data transfer	Value is loaded at the specified location	NONE
	MOV	0AH, #0ECH	Immediate	1	2	Data transfer	Value is loaded at the specified location	NONE

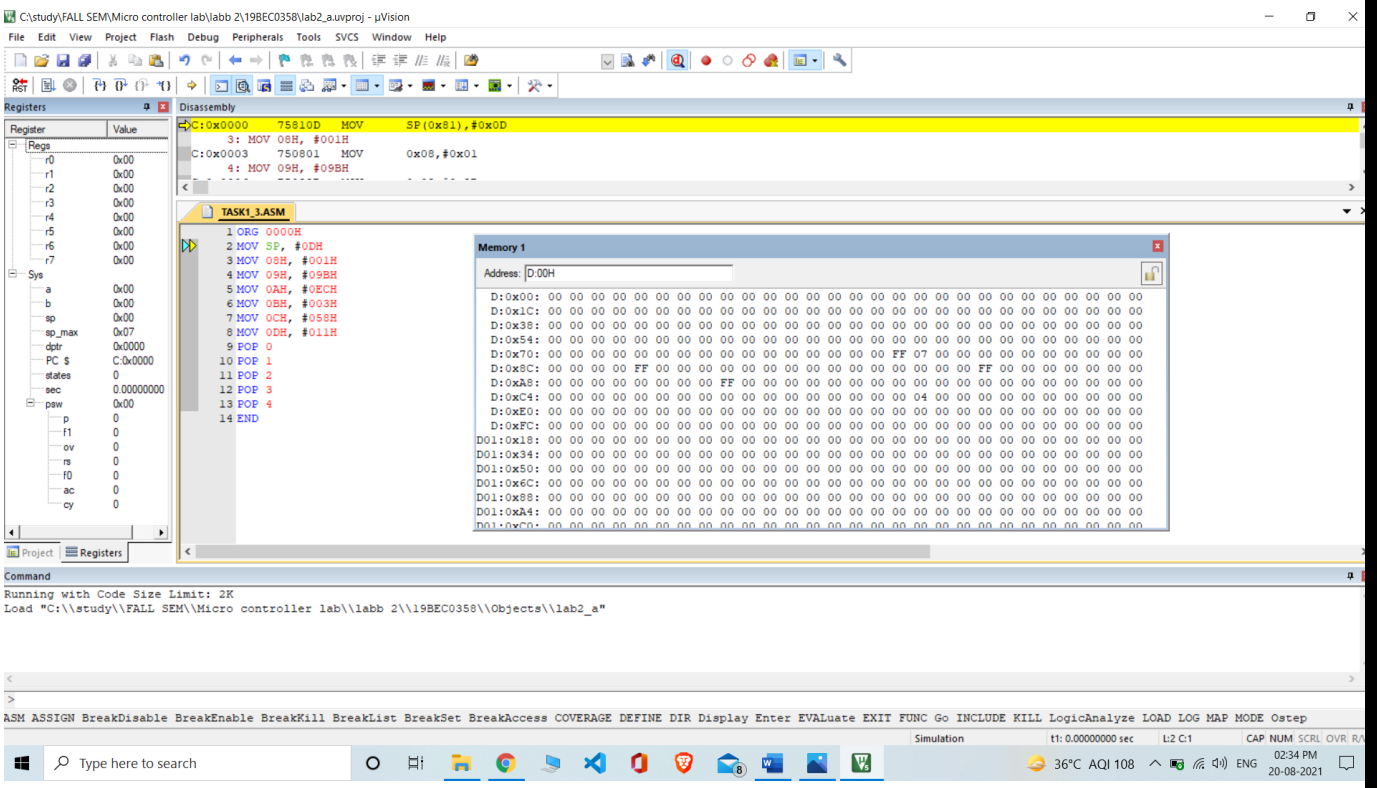
	MOV	0BH, #003H	Immediate	1	2	Data transfer	Value is loaded at the specified location	NONE
	MOV	0CH, #058H	Immediate	1	2	Data transfer	Value is loaded at the specified location	NONE
	MOV	011H, #00DH	Immediate	1	2	Data transfer	Value is loaded at the specified location	NONE
	POP	0	Direct	2	2	Data transfer	Value is popped from stack to R0	NONE
	POP	1	Direct	2	2	Data transfer	Value is popped from stack to R1	NONE
	POP	2	Direct	2	2	Data transfer	Value is popped from stack to R2	NONE
	POP	3	Direct	2	2	Data transfer	Value is popped from stack to R3	NONE
	POP	4	Direct	2	2	Data transfer	Value is popped from stack to R4	NONE
	END						Programme ends here	NONE

Output: Registers containing the Result: R0 =11, R1=58, R2 =03, R3=EC, R4 = 9B

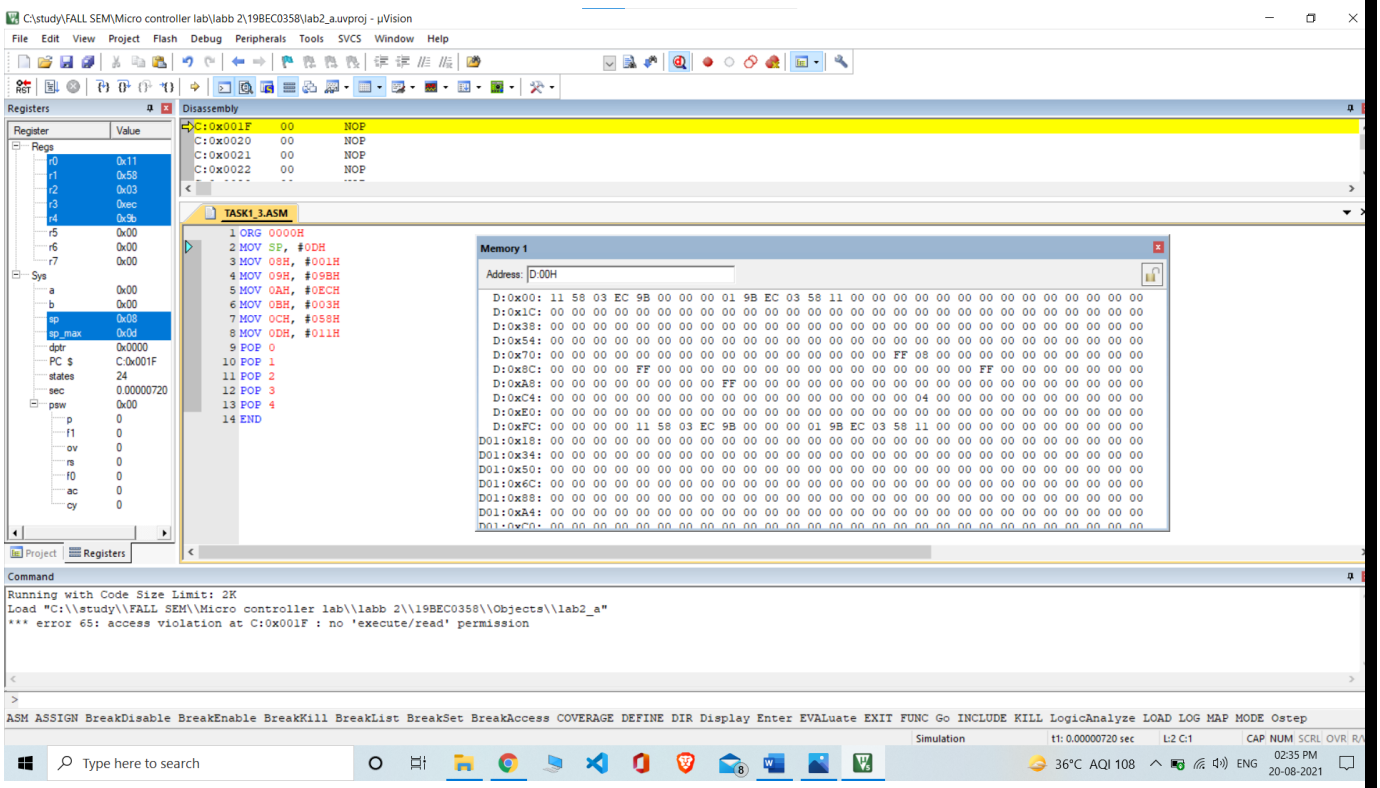
Manual Calculation: NONE

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – All are 0.

2. ABOUT THE OUTPUT VALUES IN REGISTERS – all the data from stack is Pushed back to Registers R0-R4.

Result: The 8051 ALP to POP 5 8Bit numbers is executed using keil software and the results are verified manually.

----------XXXXX----------

TASK 4)

Aim: To write an 8051 ALP to PUSH values from Registers to Stack and POP them back using keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.Load different numbers to registers from R0 -R4

2.PUSH each value to stack

3.POP each value back from stack

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000h						
	MOV	R0, #025H	Immediate	1	2	Data transfer	Value is loaded to R0	NONE
	MOV	R1, #035H	Immediate	1	2	Data transfer	Value is loaded to R1	NONE
	MOV	R2, #045H	Immediate	1	2	Data transfer	Value is loaded to R2	NONE
	MOV	R3, #055H	Immediate	1	2	Data transfer	Value is loaded to R3	NONE
	MOV	R4, #058H	Immediate	1	2	Data transfer	Value is loaded to R4	NONE
	MOV	R5, #065H	Immediate	1	2	Data transfer	Value is loaded to R5	NONE
	PUSH	0	Direct	2	2	Data	Value is	NONE

						transfer	pushed to stack	
	PUSH	1	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	2	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	3	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	4	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	PUSH	5	Direct	2	2	Data transfer	Value is pushed to stack	NONE
	POP	0	Direct	2	2	Data transfer	Value is popped from stack to R0	NONE
	POP	1	Direct	2	2	Data transfer	Value is popped from stack to R1	NONE
	POP	2	Direct	2	2	Data transfer	Value is popped from stack to R2	NONE
	POP	3	Direct	2	2	Data transfer	Value is popped from stack to R3	NONE
	POP	4	Direct	2	2	Data transfer	Value is popped from stack to R4	NONE

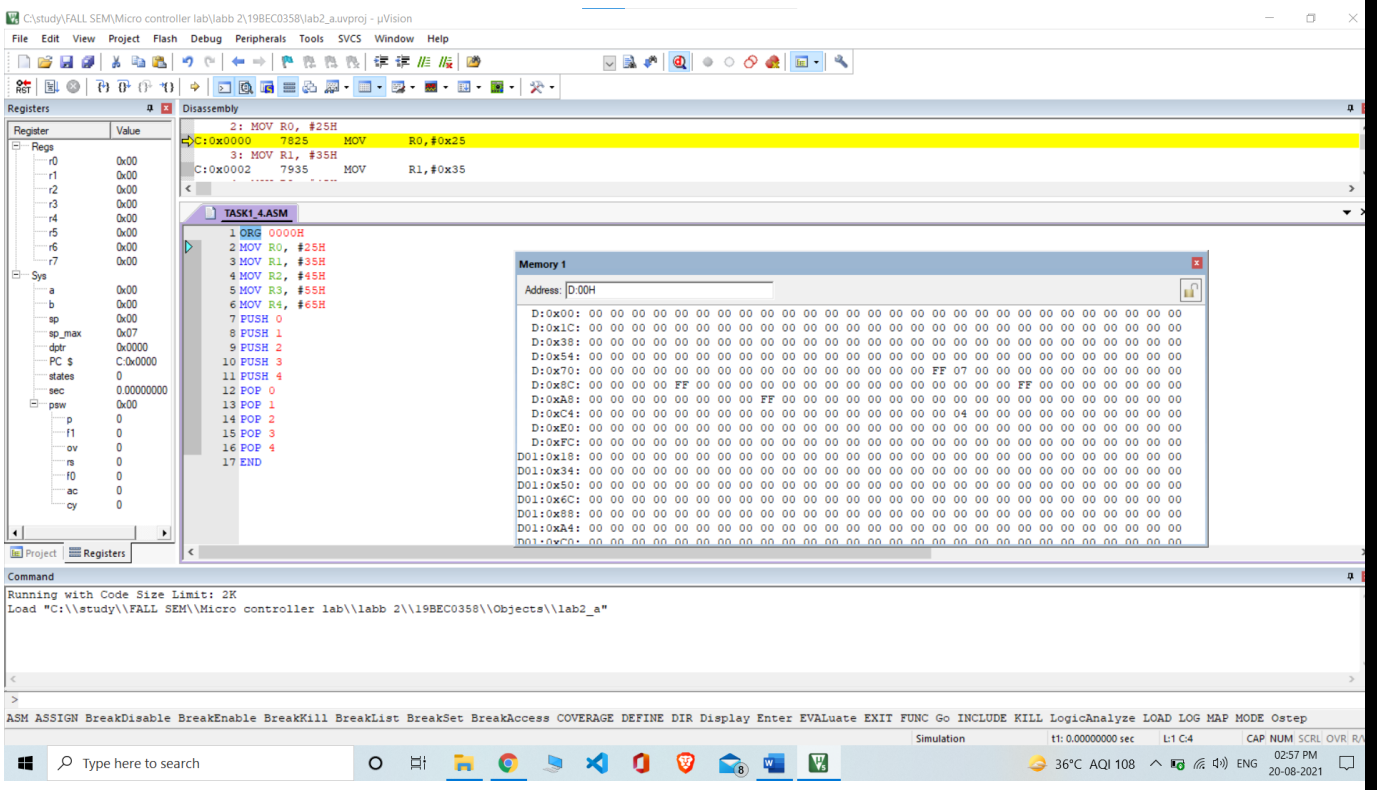
	END						Programme ends here	NONE

Output: Registers containing the Result: R0 =65, R1=55, R2 =45, R3=35, R4 = 25

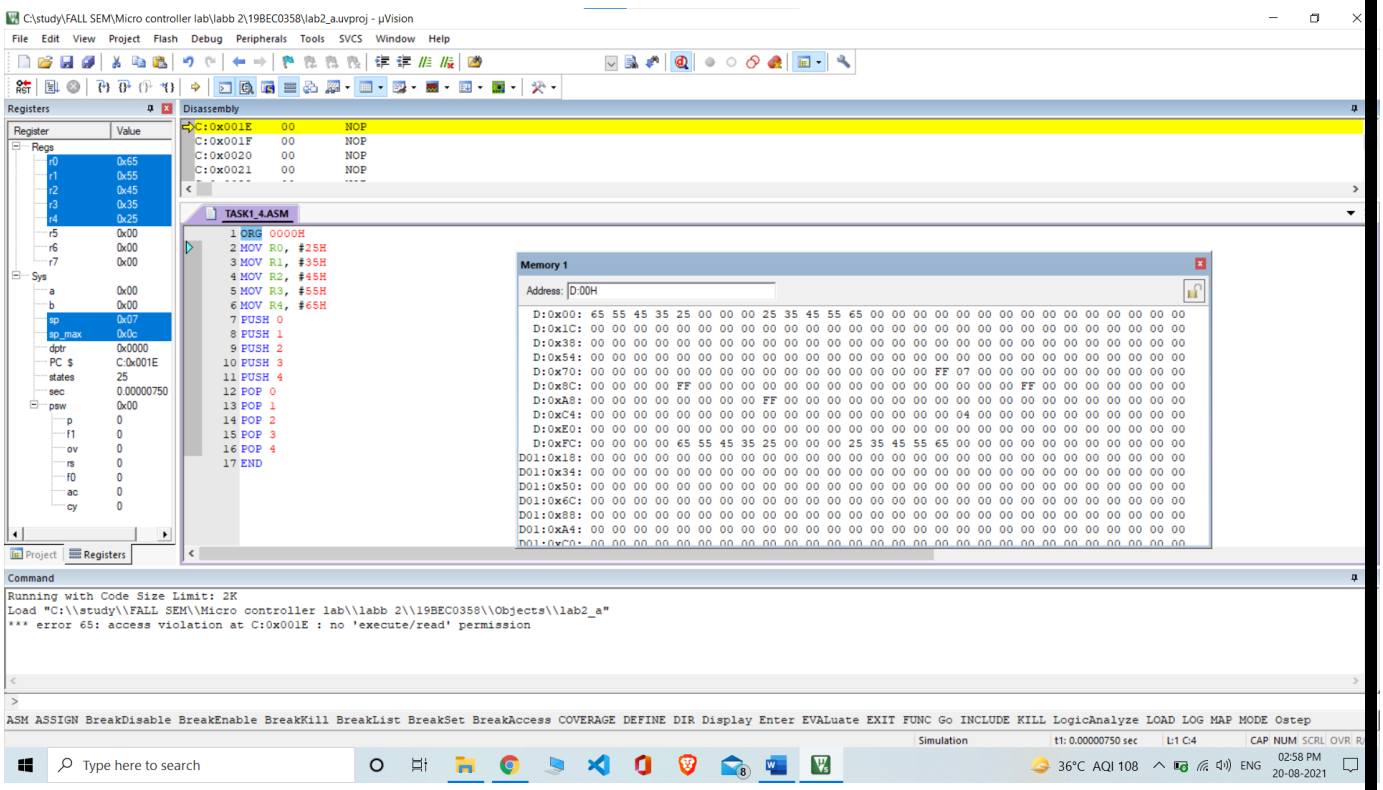
Manual Calculation: NONE

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – All are 0.

2. ABOUT THE OUTPUT VALUES IN REGISTERS – form R0 to R4 all the data is pushed to stack Memory and again POP back, so their value gets reversed.

Result: The 8051 ALP to PUSH and POP of 5 8Bit numbers is executed using Keil software and the results are verified manually.

----------XXX----------

TASK 5.a)

Aim: To write an 8051 ALP to add 10 bytes of BCD data and store the result in R2 and R3 using Keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.take 10 bytes of data and store it in DB at location C:300H

2.move data pointer to 300H location.

3. take data from DB, load it to A, add with R2 and store to R2.

4.check for decimal adjust accumulator content and increment R3 if it is present.

5. run a loop for 10 times to repeat this process.

6. finally the result is stored in R2 and R3.

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	000H						
	MOV	DPTR, #300H	Immediate	2	3	Data transfer	Data pointer is moved at the 300th location	NONE
	MOV	R0, #10H	Immediate	1	2	Data transfer	R0 is loaded with value 10	NONE
LOOP	CLR	A	Immediate	1	1	Logical	Clear value of A	NONE
	MOVC	A, @A+DPTR	Indexed	2	1	Data transfer	Move location to DPTR + @A	NONE
	ADD	A, R2	Register	1	1	Arithmet	Add R2 and A and	PARITY

						ic	store in A	
	DA	A	Immediate	1	1	Arithmetic	Check for decimal adjust accumulator content for A	NONE
	JNC	NEXT	Indexed	2	2	Boolean	Jump to next if there is no carry	NONE
	INC	R3	Immediate	1	1	Arithmetic	Increment R3	NONE
NEXT	INC	DPTR	immediate	2	1	arithmetic	Increment DPTR	NONE
	MOV	R2, A	Register	1	1	Data transfer	Load the value of A into R2	NONE
	DZNZ	R0, LOOP	Indexed	2	2	Branching	Decrement and jump on non-zero	NONE
HERE	SJMP	HERE	Indexed	2	2	Branching	Short jump	NONE
	ORG	300H						NONE
	DB	22H, 43H, 23H, 34H, 31H, 77H, 91H, 33H, 43H, 07H						NONE

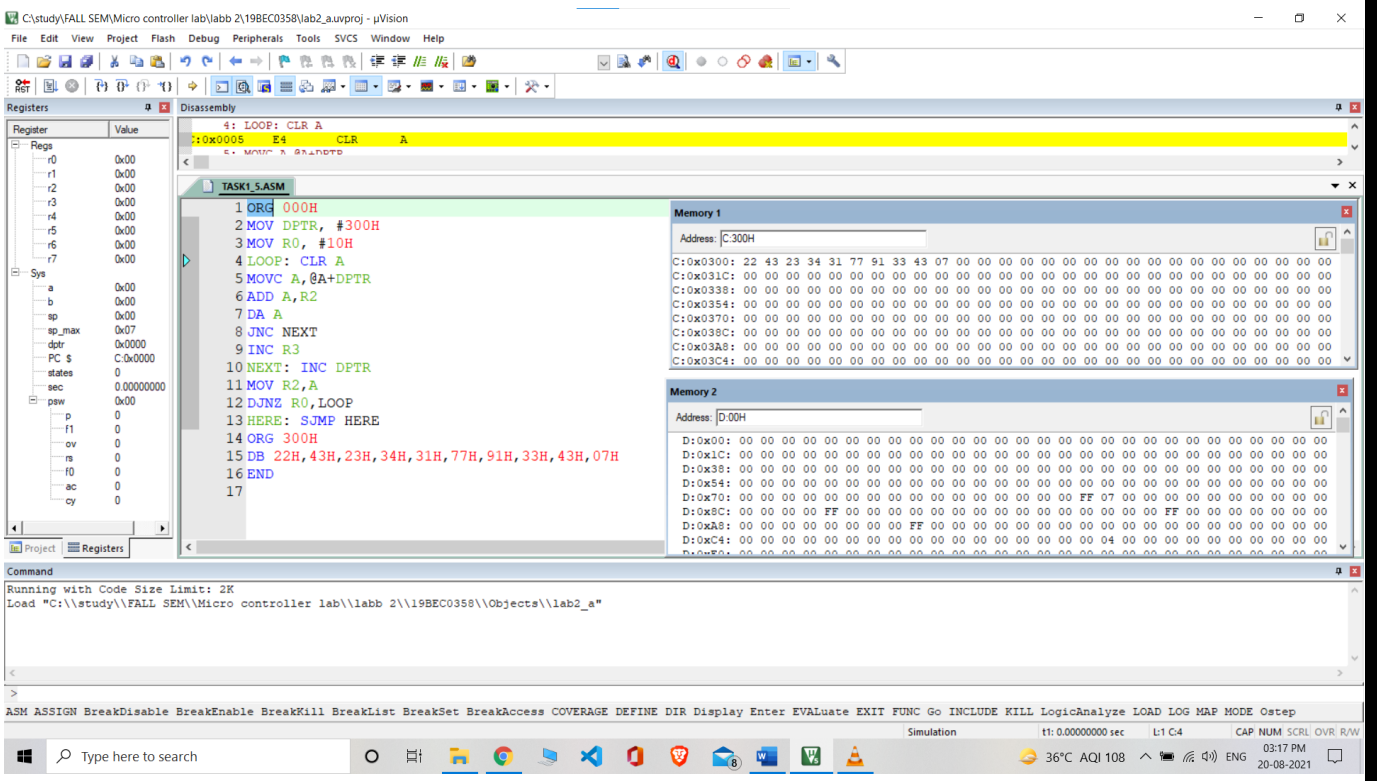
Output: Registers containing the Result: R2 =04, R3=04

Manual Calculation:

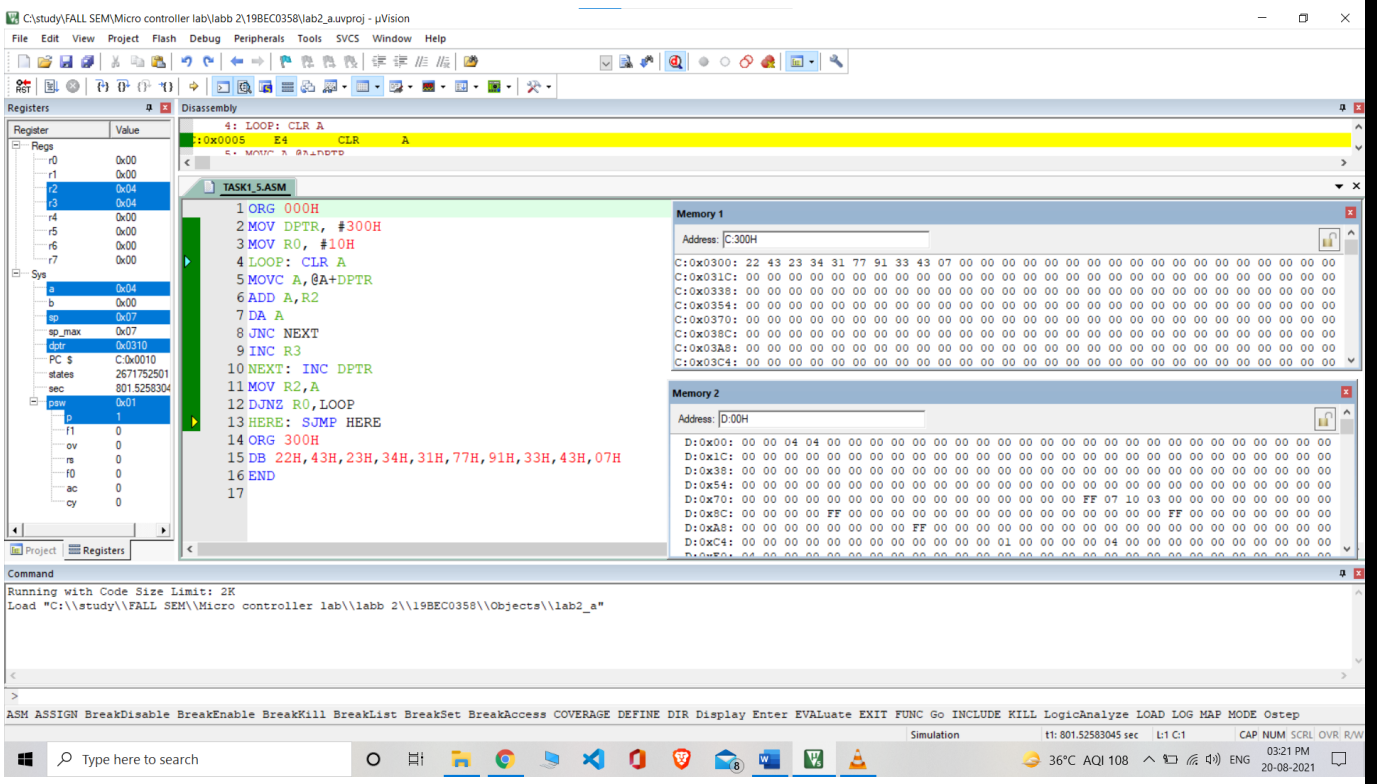
22+43+23+34+31+77+91+33+43+07= 404 in decimal

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – All are 0 except parity flag which is 1

2. ABOUT THE OUTPUT VALUES IN REGISTERS – the data is taken and stored in R2 and R3.

Result: The 8051 ALP to perform ADDITION of 10 BCD NUMBERS is executed using Keil software and the results are verified manually.

----------XXX----------

TASK 5.b)

Aim: To write an 8051 ALP to add 10 bytes of BCD data and store the result in R2 and R3 using Keil software and to verify the result manually.

Tools Required: Keil Microvision Software

Algorithm:

1.take 10 bytes of data and store it in DB at location C:300H

2.move data pointer to 300H location.

3. take data from DB, load it to A, add with R2 and store to R2.

4.check for decimal adjust accumulator content and increment R3 if it is present.

5. run a loop for 10 times to repeat this process.

6. finally the result is stored in R2 and R3.

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	000H						
	MOV	DPTR, #300H	Immediate	2	3	Data transfer	Data pointer is moved at the 300th location	NONE
	MOV	R0, #10H	Immediate	1	2	Data transfer	R0 is loaded with value 10	NONE
LOOP	CLR	A	Immediate	1	1	Logical	Clear value of A	NONE
	MOVC	A, @A+DPTR	Indexed	2	1	Data transfer	Move location to DPTR + @A	NONE
	ADD	A, R2	Register	1	1	Arithmet	Add R2 and A and	PARITY

						ic	store in A	
	DA	A	Immediate	1	1	Arithmetic	Check for decimal adjust accumulator content for A	NONE
	JNC	NEXT	Indexed	2	2	Boolean	Jump to next if there is no carry	NONE
	INC	R3	Immediate	1	1	Arithmetic	Increment R3	NONE
NEXT	INC	DPTR	immediate	2	1	arithmetic	Increment DPTR	NONE
	MOV	R2, A	Register	1	1	Data transfer	Load the value of A into R2	NONE
	DZNZ	R0, LOOP	Indexed	2	2	Branching	Decrement and jump on non-zero	NONE
HERE	SJMP	HERE	Indexed	2	2	Branching	Short jump	NONE
	ORG	300H						NONE
	DB	01H, 9BH, ECH, 03H, 58H, 00H, 00H, 00H, 00H, 00H						NONE

Output: Registers containing the Result: R2 =E3, R3=01

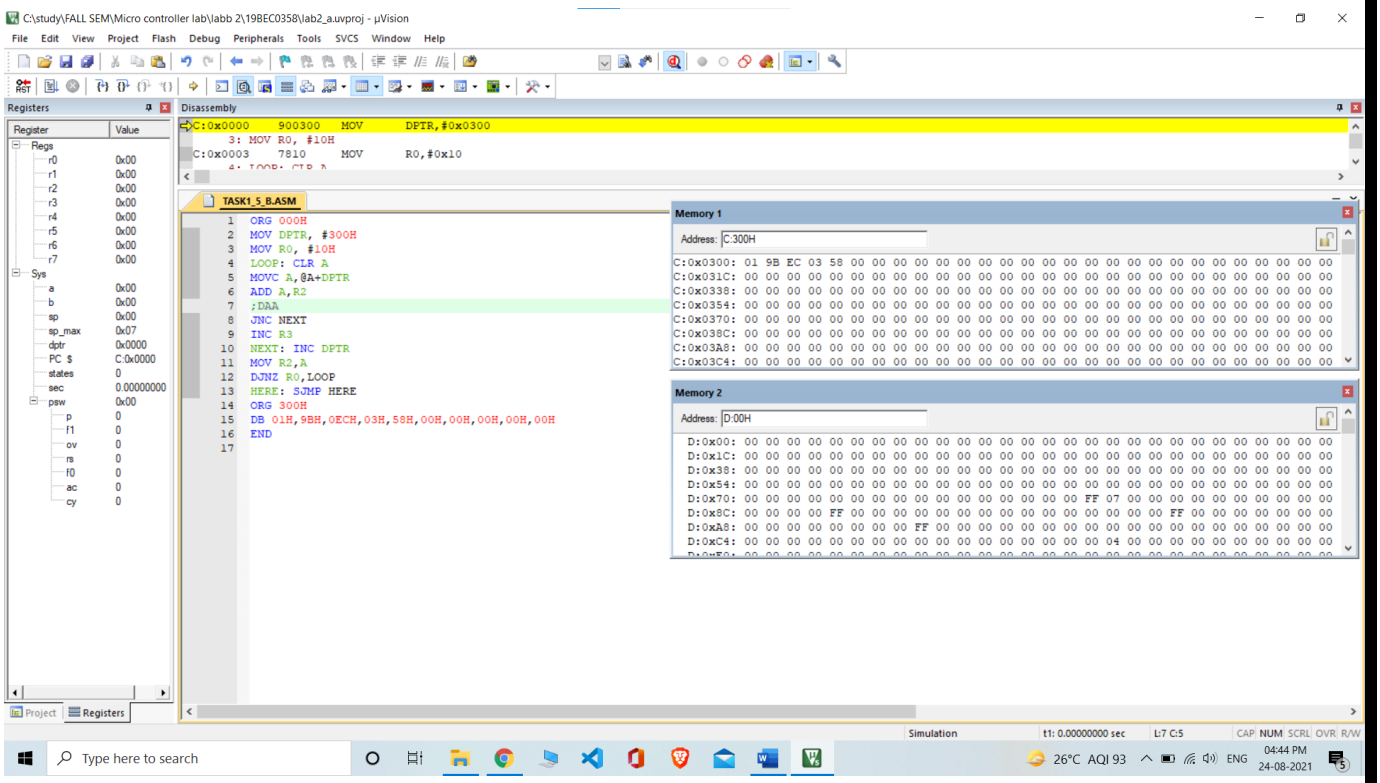
Manual Calculation: converting all numbers to decimal -

01 = 1, 9B = 155, EC = 236, 03 =3, 58 = 88

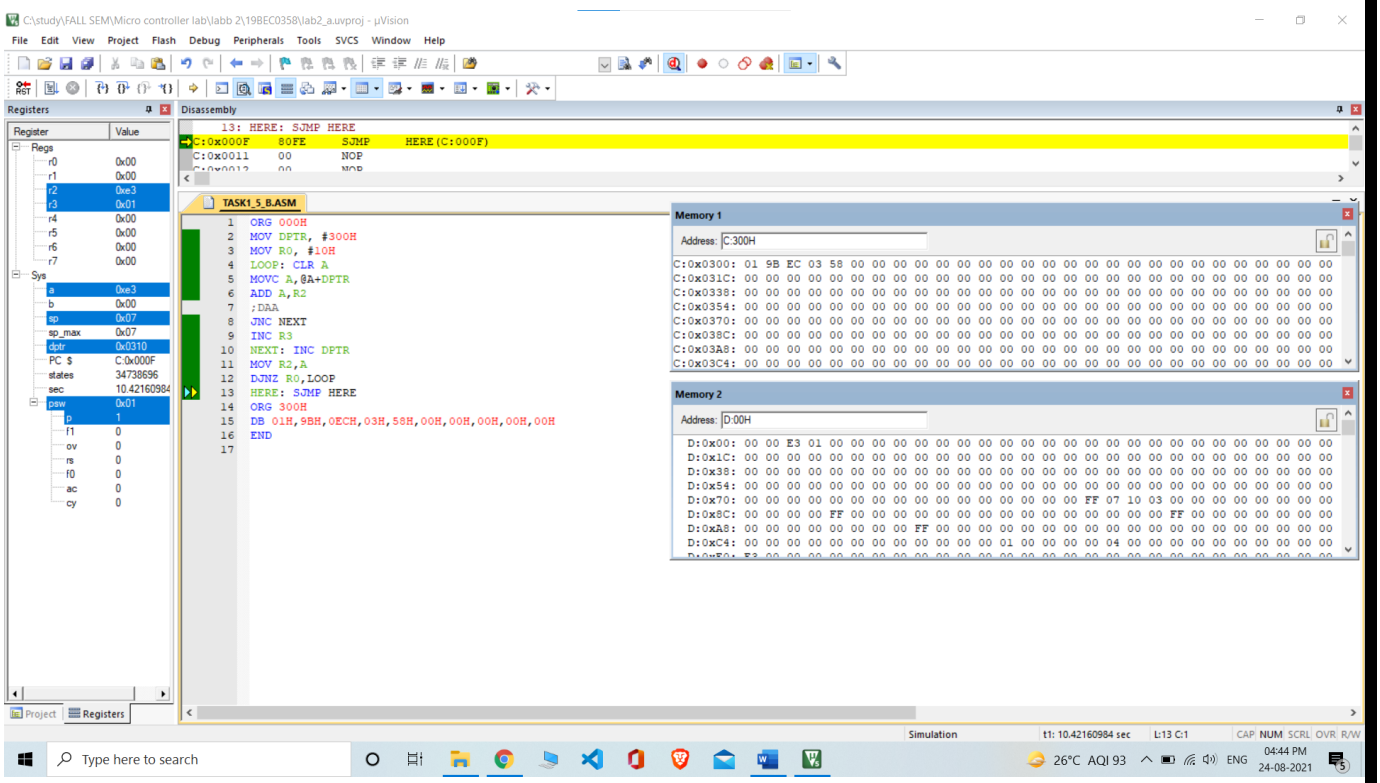
1+155+236+3+88= 483 in decimal = 1E3 in hexa decimal

Results and Observations

Program and registers before execution:



Program and registers after execution: FINAL STEP



Inferences:

1. About PSW VALUES – All are 0 except parity flag which is 1

2. ABOUT THE OUTPUT VALUES IN REGISTERS – the data is taken and stored in R3 and R2.

Result: The 8051 ALP to perform ADDITION of 10 BCD NUMBERS is executed using Keil software and the results are verified manually.

----------XXX----------