Task No.: 2

School OF Electronics Engineering
VIT, Vellore

Reg.No	17BEC0656		
Student Name	SPARSH ARYA		
Course Code	ECE3003	**Slot & Semester**	L49+L50, FALL ~2019-20
Course Name	Microcontroller and its applications		
Program Title	LAB TASK 2		
Date of Exp.	31.8.19	**Date of Submission**	31.8.19
Faculty	A.Karthikeyan		

Submission:
Mail Id :akece3003e1tt724@gmail.com , akece3003e2tt724@gmail.com
Mail Subject : REG.Number_L__+L__FALL_19_20
File Name and format : Reg.number.docx (doc) - **only**

MAIL YOUR, DIGITAL ASSIGNMENT , LAB PROGRAMS TO THE ABOVE SAID MAIL ID

And also LOAD YOUR DOCUMENTS BEFORE THE DEAD LINE ON THE INTRANET

Q.Write a program to transfer a string of data from code space starting at address 200H to RAM locations starting at 40H. The data is as shown below:

0200H:DB "VIT UNIVERSITY"

Using the simulator, single-step through the program and examine the data transfer and registers.

Aim:

To transfer string "VIT UNIVERSITY" from ROM (address 200H) to RAM locations starting at 40H

Tools Required

Keil software

Algorithm

1. Move the value 200H to data pointer.
2. Store values from the ROM location of 200H as ascii " VIT UNIVERSITY".
3. Store 0EH in R0 of bank 0. This is done to keep track of the number of values being transferred at each iteration.
4. Store value of 40H in R0 of bank 0.
5. Clear the accumulator and store the value at 200H in A.
6. Move the values stored in A to 40H, by indirect addressing mode through R0.
7. Increment R0 and datapointer.
8. Decrement R1 and run the loop till the value stored in R1 is zero.

Program:

```
ORG 0000H
MOV DPTR,#200H
MOV R1,#0EH
MOV R0,#40H
LOOP: CLR A
MOVC A,@ A+DPTR
MOV @ R0,A
INC R0
INC DPTR
DJNZ R1,LOOP
```

```
HERE: SJMP HERE
ORG 200H
DB "VIT UNIVERSITY"
END
```

Memory Address	Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type OF Instruction	Comments	Flags getting affected by the Instruction.
-	-	ORG	0000H	-	-	-	Pseudo instruction	Code stored from 0000H	-
0000H		MOV	DPTR, #200H	Immediate	2	3	Data transfer	Source address stored in DPTR	-
0003H		MOV	R1,#0EH	Immediate	1	2	Data transfer	Loop counter initialized	-
0005H		MOV	R0,#40H	Immediate	1	2	Data transfer	Destination address	-
0007H	LOOP	CLR	A	-	1	1	Arithmetic	Clear a	-
0008H		MOVC	A, @A+DPTR	Indexed	2	1	Data transfer	Transfer source data to accumulator	-
0009H		MOV	@R0,A	Register indirect	1	1	Data transfer	Move data from accumulator to RAM address in r0	-
000AH		INC	R0	-	1	1	Arithmetic	Change RAM address by 1	-
000BH		INC	DPTR	-	2	1	Arithmetic	Change ROM address by 1	-
000CH		DJNZ	R1.LOOP		2	2	Branch	Jump to loop on non-zero as	-

								looping action	
000EH	**HERE**	**SJMP**	**HERE**	**-**	**2**	**2**	**Branch**	**Infinite loop**	**-**
		ORG	**200H**	**-**	**-**	**-**	**Pseudo**	**To store data in ROM**	**-**
									-
		DB	**“VIT UNIVERSITY”**	**-**	**-**	**-** **-**	**Pseudo**	**Input storing**	**-**
		END	**-**	**-**	**-**	**-**	**Pseudo**	**End OF program**	**-**

Output :

Registers containing the result:

R0 : 04EH

A: 59H(“Y”)

DPTR: 020EH

Register OF 40H: V

Register OF 41H: I

Register OF 42H: T

Register OF 43H: “ “

Register OF 44H: U

Register OF 45H: N

Register OF 46H: I

Register OF 47H: V

Register OF 48H: E

Register OF 49H: R

Register OF 4AH: S

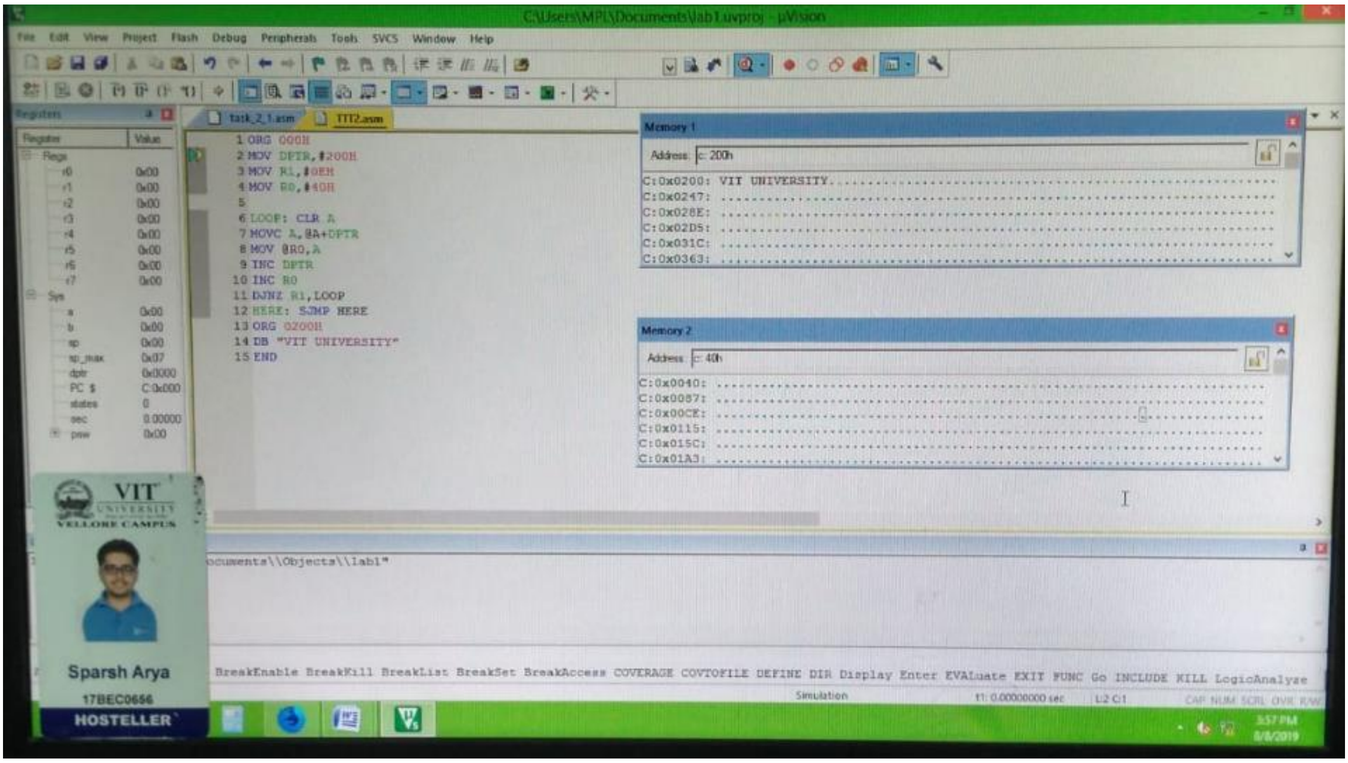
Register OF 4BH: I

Register OF 4CH: T

Register OF 4DH: Y

Results and Observations

Print Screen of the Program and Registers before execution:


C:\Users\MPL\Documents\lab1.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
task_2_1.asm
TTT2.asm
Memory 1
Memory 2
VIT UNIVERSITY VELLORE CAMPUS
Sparsh Arya
17BEC0656
HOSTELLER

```
ORG 000H
MOV DPTR,#200H
MOV R1,#0EH
MOV R0,#40H

LOOP: CLR A
MOVC A,@A+DPTR
MOV @R0,A
INC DPTR
INC R0
DJNZ R1,LOOP
HERE: SJMP HERE
ORG 0200H
DB "VIT UNIVERSITY"
END
```

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC $	C:0x000
states	0
sec	0.00000
psw	0x00

Memory 1 — Address: c: 200h

```
C:0x0200: VIT UNIVERSITY.............................................................
C:0x0247: ...................................................................
C:0x028E: ...................................................................
C:0x02D5: ...................................................................
C:0x031C: ...................................................................
C:0x0363: ...................................................................
```

Memory 2 — Address: c: 40h

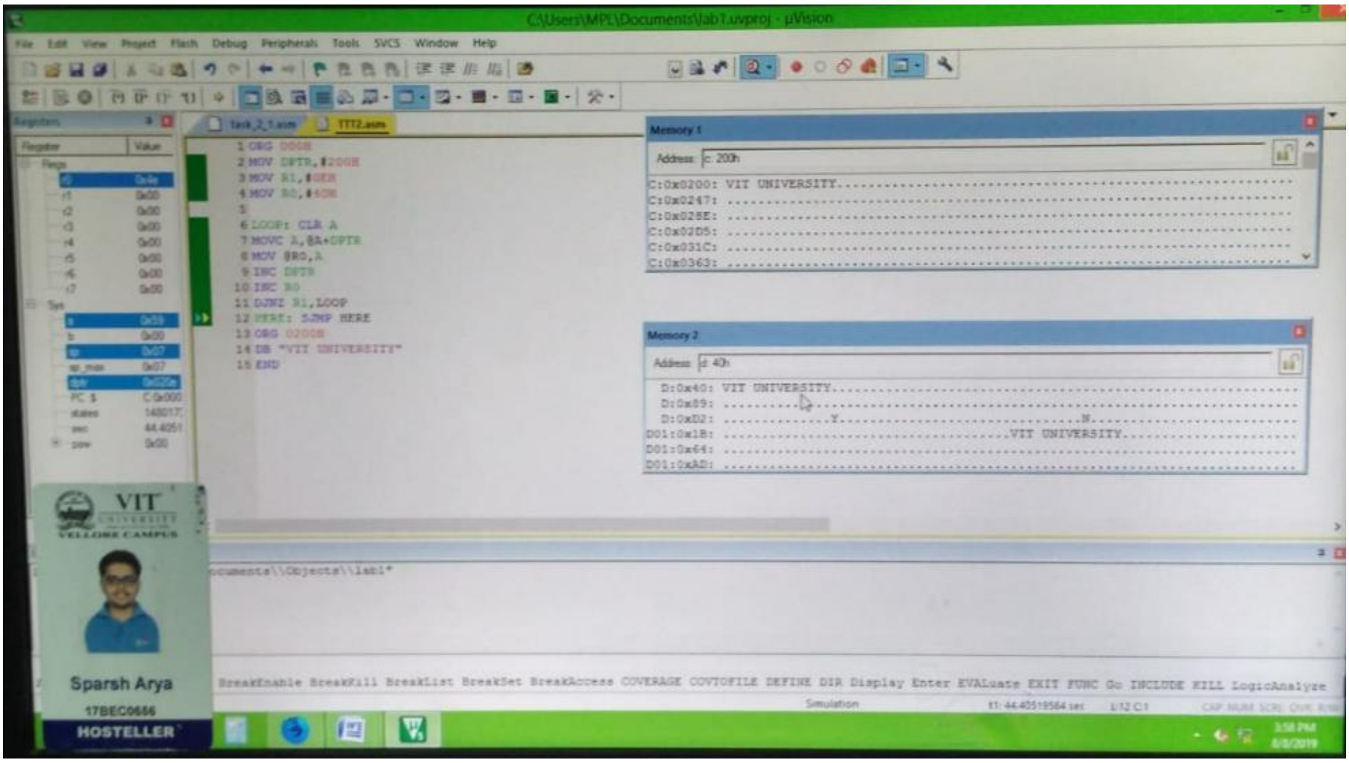
```
C:0x0040: ...................................................................
C:0x0087: ...................................................................
C:0x00CE: ...................................................................
C:0x0115: ...................................................................
C:0x015C: ...................................................................
C:0x01A3: ...................................................................
```

```
ocuments\\Objects\\lab1"
```

```
BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze
```

Simulation t1: 0.00000000 sec L:2 C:1 CAP NUM SCRL OVR R/W

Print Screen of the program and registers after execution:


C:\Users\MPL\Documents\lab1.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
task_2_1.asm
TTT2.asm
Memory 1
Memory 2
VIT UNIVERSITY VELLORE CAMPUS
Sparsh Arya
17BEC0656
HOSTELLER

```
ORG 0000H
MOV DPTR,#200H
MOV R1,#0EH
MOV R0,#40H

LOOP: CLR A
MOVC A,@A+DPTR
MOV @R0,A
INC DPTR
INC R0
DJNZ R1,LOOP
HERE: SJMP HERE
ORG 0200H
DB "VIT UNIVERSITY"
END
```

```
Address: c: 200h
C:0x0200: VIT UNIVERSITY.......................................................
C:0x0247: .....................................................................
C:0x028E: .....................................................................
C:0x02D5: .....................................................................
C:0x031C: .....................................................................
C:0x0363: .....................................................................
```

```
Address: d: 40h
  D:0x40: VIT UNIVERSITY.......................................................
  D:0x89: .....................................................................
  D:0xD2: ..............Y...........................N..........................
D01:0x1B: ...............................VIT UNIVERSITY........................
D01:0x64: .....................................................................
D01:0xAD: .....................................................................
```

Result:

The 8051 ALP to move a string of data from ROM to RAM is executed using keil software and the results are verified manually.

Q.Add the following subroutine to the program 1, single-step through the subroutine and examine the RAM locations. After data has been transferred from ROM space into RAM, the subroutine should copy the data from RAM locations starting at 40H to RAM locations starting at 60H.

Aim:

To transfer string “VIT UNIVERSITY” from ROM (address 200H) to RAM locations starting at 40H with additional subroutine as specified.

Tools Required:

Keil software

Algorithm:

1. Move the value 200H to data pointer.
2. Store values from the ROM location of 200H as ascii “ VIT UNIVERSITY”.
3. Store 0EH in R0 of bank 0. This is done to keep track of the number of values being transferred at each iteration.
4. Store value of 40H in R0 of bank 0.
5. Clear the accumulator and store the value at 200H in A.
6. Move the values stored in A to 40H, by indirect addressing mode through R0.
7. Increment R0 and datapointer.
8. Decrement R1 and run the loop till the value stored in R1 is zero.
9. Move value of 0EH to R2 of bank 0 and R1(bank 0) as 40H.
10. Move 60H to R0 of bank 0.
11. Use indirect adressing mode to move all values to accumulator and then to 60H.
12. Increment R0 and R1.
13. Decrement R2 and run the loop til R2 is zero.

Program:

```
ORG 0000H

MOV DPTR,#200H

MOV R1,#0EH
MOV R0,#40H
LOOP: CLR A
MOVC A,@ A+DPTR
MOV @ R0,A
INC R0
INC DPTR
DJNZ R1,LOOP
MOV R2,#0EH
MOV R1,#40H
MOV R0,#60H
AGAIN: CLR A
MOV A,@R1
MOV @R0,A
INC R1
INC R0
DJNZ R2,AGAIN
HERE: SJMP HERE
ORG 200H
DB "VIT UNIVERSITY"
END
```

Memory Address	**Label**	**Mnemonics**	**Operands**	**addressing mode used**	**Machine cycle Required**	**Memory Byte Required**	**Type OF Instruction**	**Comments**	**Flags getting affected by the Instruction.**
-	**-**	**ORG**	**0000H**	**-**	**-**	**-**	**Pseudo instruction**	**Code stored from 0000H**	**-**
0000H		**MOV**	**DPTR, #200H**	**Immediate**	**2**	**3**	**Data transfer**	**Source address stored in DPTR**	**-**
0003H		**MOV**	**R1,#0EH**	**Immediate**	**1**	**2**	**Data transfer**	**Loop counter initialized**	**-**
0005H		**MOV**	**R0,#40H**	**Immediate**	**1**	**2**	**Data transfer**	**Destination address**	**-**
0007H	**LOOP**	**CLR**	**A**	**-**	**1**	**1**	**ARITHMETIC**	**CLEAR A**	**-**
0008H		**MOVC**	**A,**	**Indexed**	**2**	**1**	**Data transfer**	**Transfer source data**	**-**

			@A+DPTR					to Accumulator	
0009H		MOV	@R0,A	Register indirect	1	1	Data transfer	Move data from ROM Accumulator to RAM address in r0	-
000AH		INC	R0	-	1	1	Arithmetic	Change RAM address by 1	-
000BH		INC	DPTR	-	2	1	Arithmetic	Change ROM address by 1	-
000CH		DJNZ	R1.LOOP		2	2	Branch	Jump to loop on non-zero as looping action	-
000EH		MOV	R,#0EH	Immediate	1	2	Data transfer	Initialising loop counter	
0010H		MOV	R1,#40H	Immediate	1	2	Data transfer	Storing source address	
0012H		MOV	R0,#60H	Immediate	1	2	Data transfer	Storing destination address	
0014H	AGAIN	CLR	A	-	1	1	Arithmetic	Clear value OF Accumulato r	
0015H		MOV	A,@R1	Register indirect	1	1	Data transfer	Data sent to Accumulato r	-
0016H		MOV	@R0,A	Register indirect	1	1	Data transfer	Data transferred to destination RAM address	-
0017H		INC	R1	-	1	1	Arithmetic	Change source address by	

								1	
0018H		**INC**	**R0**	**-**	**1**	**1**	**Arithmetic**	**Change destination address by 1**	
0019H		**DJNZ**	**R2,AGAIN**		**2**	**2**	**Branch**	**Jump to loop for repetition**	
0021H	**HERE**	**SJMP**	**HERE**	**-**	**2**	**2**	**Branch**	**Infinite loop**	**-**
		ORG	**200H**	**-**	**-**	**-**	**Pseudo**	**To store data in rom**	**-**
									-
		DB	**“VIT UNIVERSITY”**	**-**	**-**	**-** **-**	**Pseudo**	**Input storing**	**-**
		END	**-**	**-**	**-**	**-**	**Pseudo**	**End of program**	**-**

Output :

Registers containing the result:

R0 : 06EH

R1 : 04EH

A: 59H(“Y”)

DPTR: 020EH

Register OF 60H: V

Register OF 61H: I

Register OF 62H: T

Register OF 63H: “ “

Register OF 64H: U

Register OF 65H: N

Register OF 66H: I

Register of 67H: V

Register of 68H: E

Register of 69H: R

Register OF 6AH: S

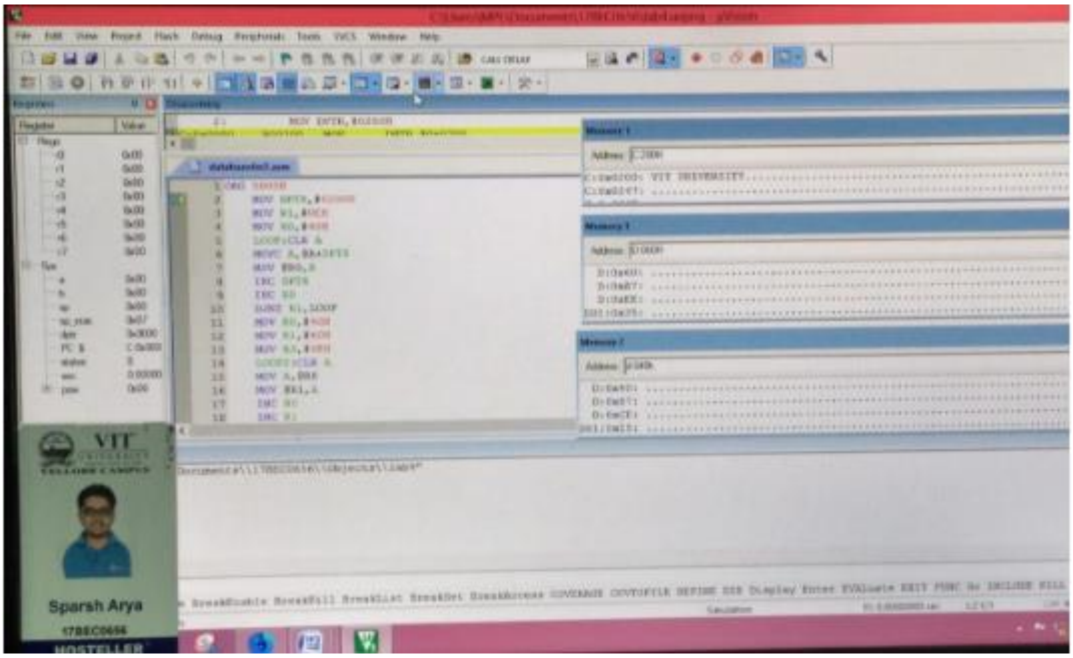
Register OF 6BH: I

Register OF 6CH: T

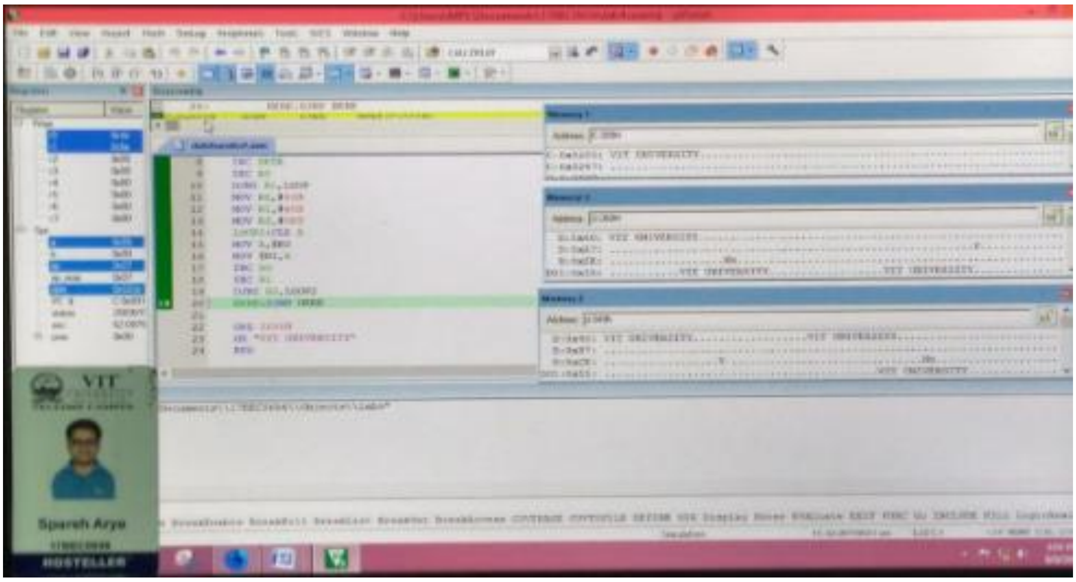
Register OF 6DH: Y

Results and Observations

Print Screen OF the Program and Registers before execution:



Print Screen OF the Program and Registers after execution:



Result:

The 8051 ALP to move a string of data from ROM to RAM and further within RAM is executed using keil software and the results are verified manually.