**COMPUTER ORGANISATION**

**EXPERIMENT 4  
ASSIGNMENT 1 (QUES 4)**

Aakash Deep (2015001)

Anannya Uberoi (2015014)

Akarsha Sehwag (2015010)

YS Ramya (2015117)

Sarthak Jindal (2015169)

# **OBJECTIVE**

To explain the use of stack and stack addressing with the help of an assembly language program and show how various programming elements are assigned memory, registers in direct and indirect addressing modes.

# **SOFTWARE REQUIREMENTS**

Keil Version 5.20.0.39

# **HARDWARE REQUIREMENTS**

None

**DESCRIPTION**

Stack:

ORG 0000

MOV B,#30H

MOV R3,#20H

PUSH 03 ;Push R3 on the stack

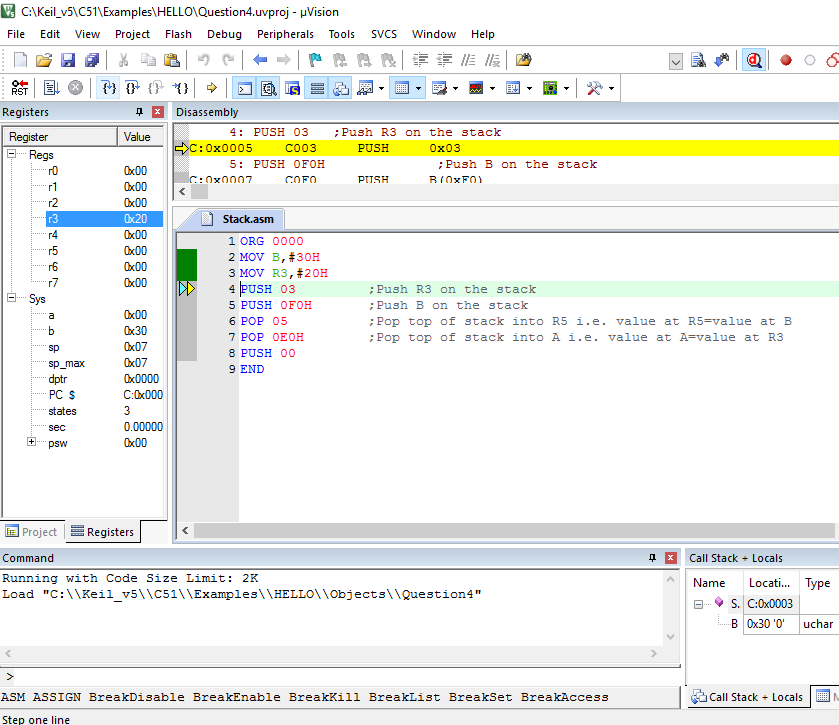
PUSH 0F0H ;Push B on the stack

POP 05 ;Pop top of stack into R5 i.e. value at R5=value at B

POP 0E0H ;Pop top of stack into A i.e. value at A=value at R3

PUSH 00

END



Direct Addressing:

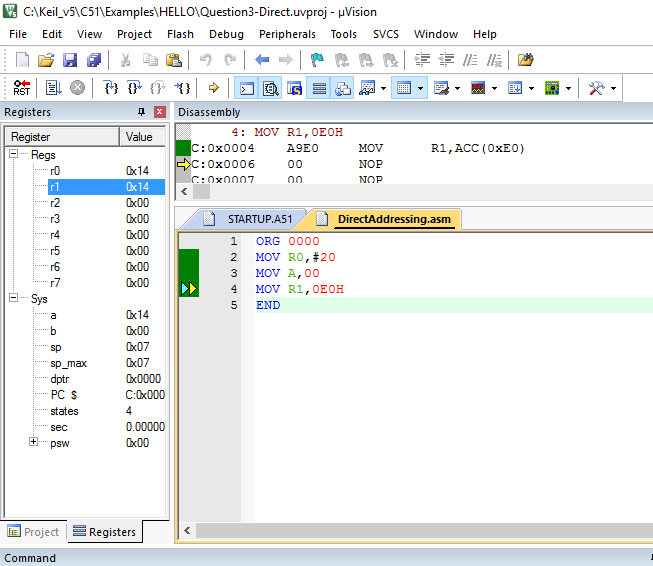
ORG 0000

MOV R0,#20

MOV A, 00H

MOV R1,0E0H

END



Indirect Addressing:

ORG 0000

MOV R2,#20H ;Direct addressing

MOV R0,#2 ;Direct addressing

MOV A,@R0 ;Indirect addressing ; the value 20H is stored in A now

END

**BLOCK DIAGRAM / SCHEMATIC DIAGRAM**

None

**COMPONENTS**

None

**RESULT**

The use of stack and stack addressing with the help of a C program/Assembly language program and show how various programming elements are assigned memory, registers in direct and indirect addressing modes have been shown.

**CONCLUSION**

We learnt how to use push and pop operations using stack. We also learnt that in direct addressing the value at the source address is copied on the destination register.. Whereas, in indirect addressing the value of the address which the register points to is stored in the destination register.

**REMARKS**

Different programs should be written and tested using assembly/C language for better understanding of the tool.