OC EXPERIMENT LAB 7

TITLE: Writing 8051 assembly language program of basic arithmetic operations

NAME: Yash Gupta

ROLL NO: S20200010234

OBSERVATION: In this lab I learnt how to write 8051 assembly language programs in keil software of basic arithmetic operations using different addressing modes.

Q1. Write an 8051-assembly language program to perform the following operations

- addition

- subtraction

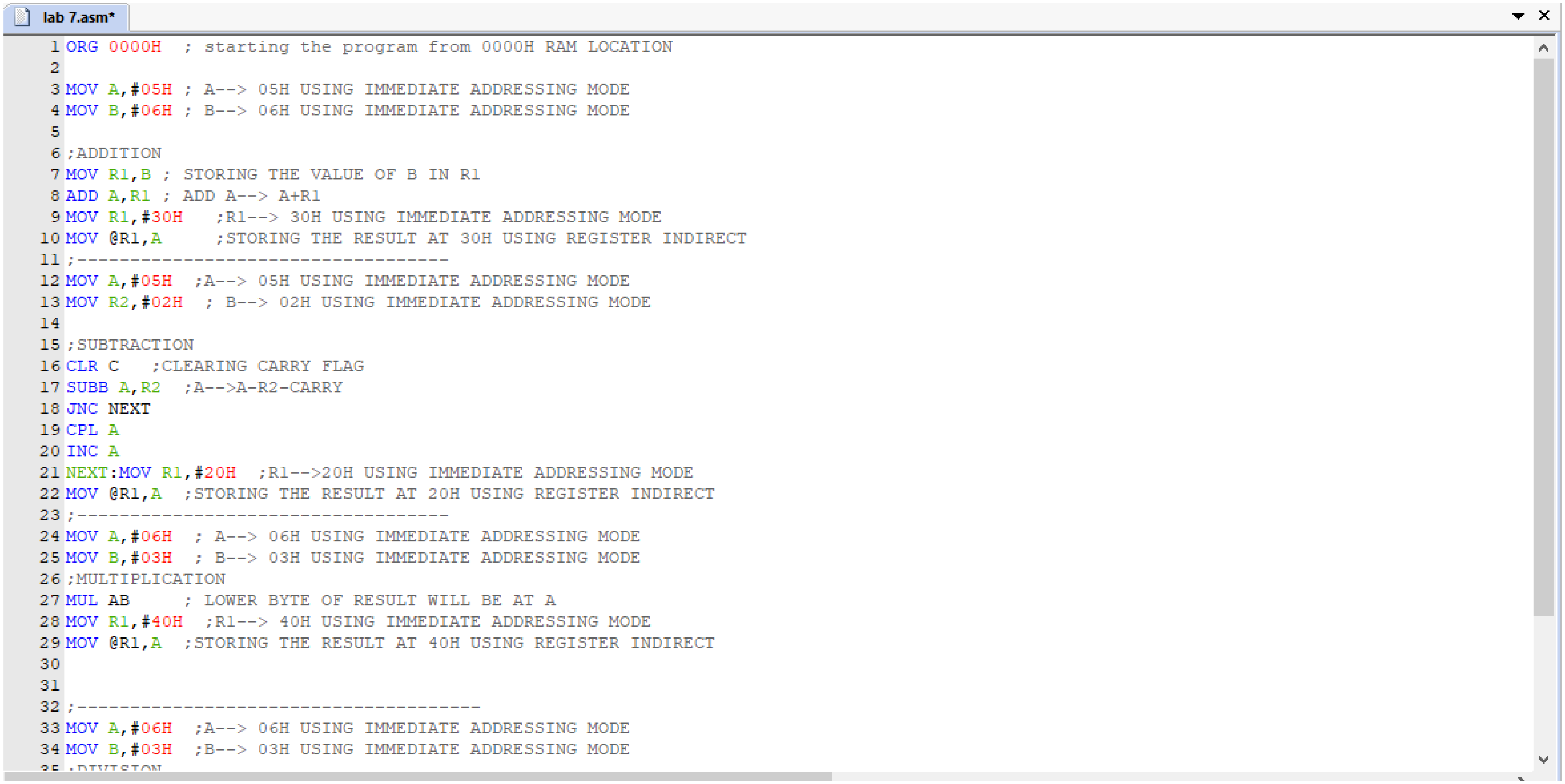
- division

- multiplication

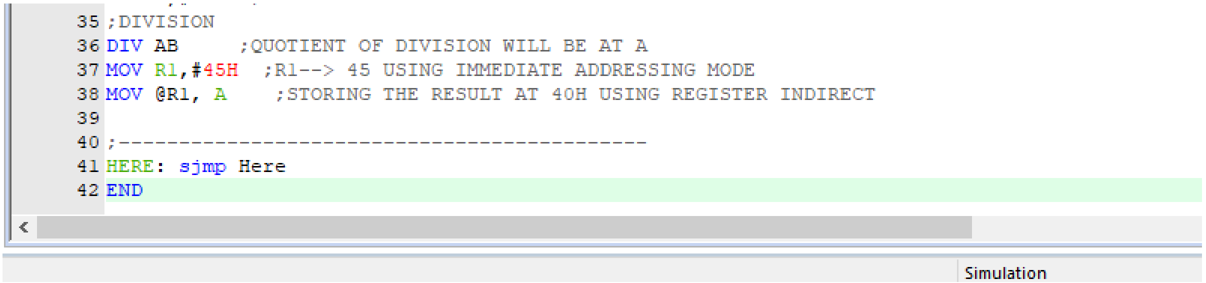
between two numbers using the concept of different addressing modes

and also illustrates the flags (AC, CY, OV, PF).

PROGRAM:



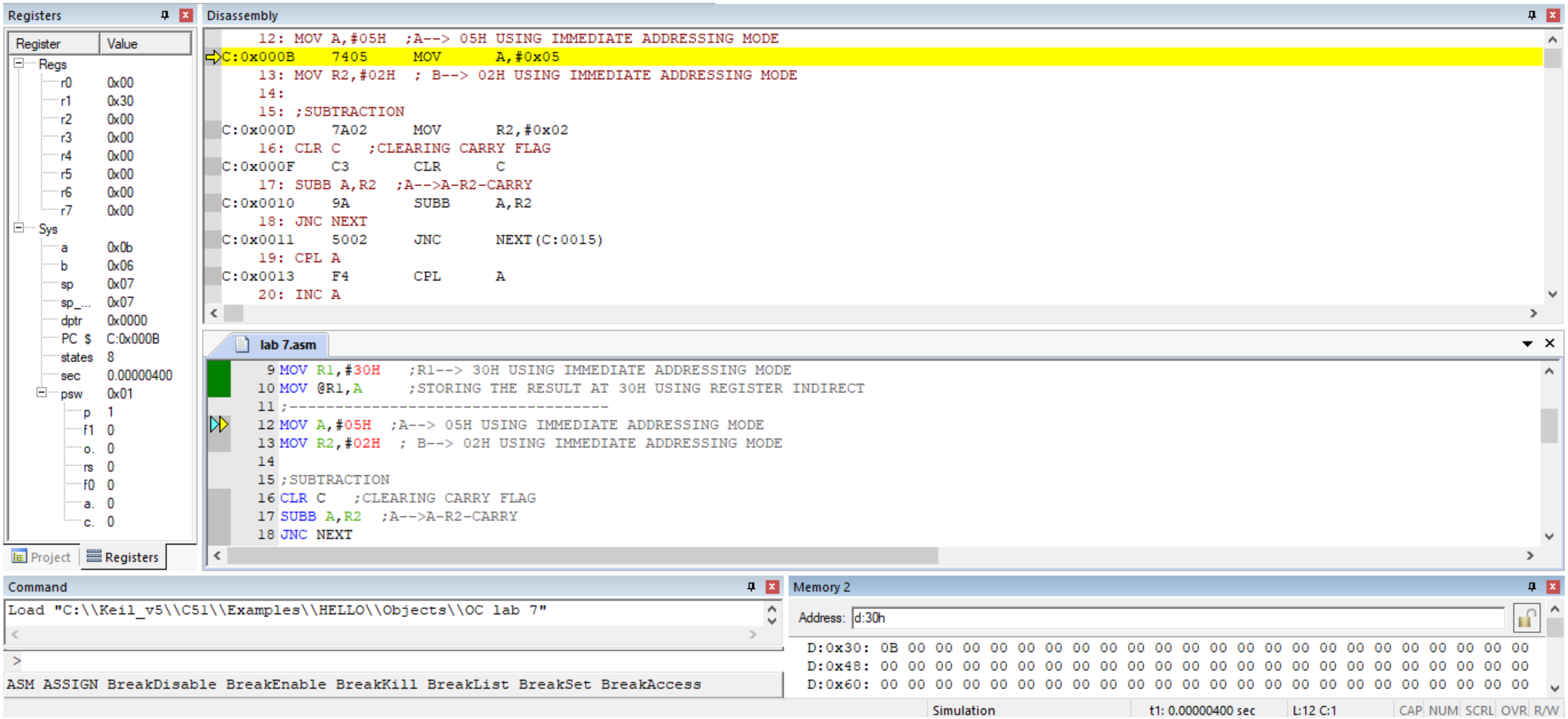
```
ORG 0000H  ; starting the program from 0000H RAM LOCATION

MOV A,#05H ; A--> 05H USING IMMEDIATE ADDRESSING MODE
MOV B,#06H ; B--> 06H USING IMMEDIATE ADDRESSING MODE

;ADDITION
MOV R1,B ; STORING THE VALUE OF B IN R1
ADD A,R1 ; ADD A--> A+R1
MOV R1,#30H   ;R1--> 30H USING IMMEDIATE ADDRESSING MODE
MOV @R1,A     ;STORING THE RESULT AT 30H USING REGISTER INDIRECT
;-----------------------------------
MOV A,#05H  ;A--> 05H USING IMMEDIATE ADDRESSING MODE
MOV R2,#02H  ; B--> 02H USING IMMEDIATE ADDRESSING MODE

;SUBTRACTION
CLR C   ;CLEARING CARRY FLAG
SUBB A,R2  ;A-->A-R2-CARRY
JNC NEXT
CPL A
INC A
NEXT:MOV R1,#20H  ;R1-->20H USING IMMEDIATE ADDRESSING MODE
MOV @R1,A  ;STORING THE RESULT AT 20H USING REGISTER INDIRECT
;---------------------------------
MOV A,#06H  ; A--> 06H USING IMMEDIATE ADDRESSING MODE
MOV B,#03H  ; B--> 03H USING IMMEDIATE ADDRESSING MODE
;MULTIPLICATION
MUL AB     ; LOWER BYTE OF RESULT WILL BE AT A
MOV R1,#40H  ;R1--> 40H USING IMMEDIATE ADDRESSING MODE
MOV @R1,A  ;STORING THE RESULT AT 40H USING REGISTER INDIRECT


;--------------------------------------
MOV A,#06H  ;A--> 06H USING IMMEDIATE ADDRESSING MODE
MOV B,#03H  ;B--> 03H USING IMMEDIATE ADDRESSING MODE
;DIVISION
```



```
;DIVISION
DIV AB      ;QUOTIENT OF DIVISION WILL BE AT A
MOV R1,#45H   ;R1--> 45 USING IMMEDIATE ADDRESSING MODE
MOV @R1, A     ;STORING THE RESULT AT 40H USING REGISTER INDIRECT

;---------------------------------------------
HERE: sjmp Here
END
```

EMULATING:

After addition:



Register	Value
Regs	
r0	0x00
r1	0x30
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x0b
b	0x06
sp	0x07
sp_...	0x07
dptr	0x0000
PC $	C:0x000B
states	8
sec	0.00000400
psw	0x01
p	1
f1	0
o.	0
rs	0
f0	0
a.	0
c.	0

```
    12: MOV A,#05H  ;A--> 05H USING IMMEDIATE ADDRESSING MODE
C:0x000B    7405     MOV      A,#0x05
    13: MOV R2,#02H  ; B--> 02H USING IMMEDIATE ADDRESSING MODE
    14:
    15: ;SUBTRACTION
C:0x000D    7A02     MOV      R2,#0x02
    16: CLR C   ;CLEARING CARRY FLAG
C:0x000F    C3       CLR      C
    17: SUBB A,R2  ;A-->A-R2-CARRY
C:0x0010    9A       SUBB     A,R2
    18: JNC NEXT
C:0x0011    5002     JNC      NEXT(C:0015)
    19: CPL A
C:0x0013    F4       CPL      A
    20: INC A
```

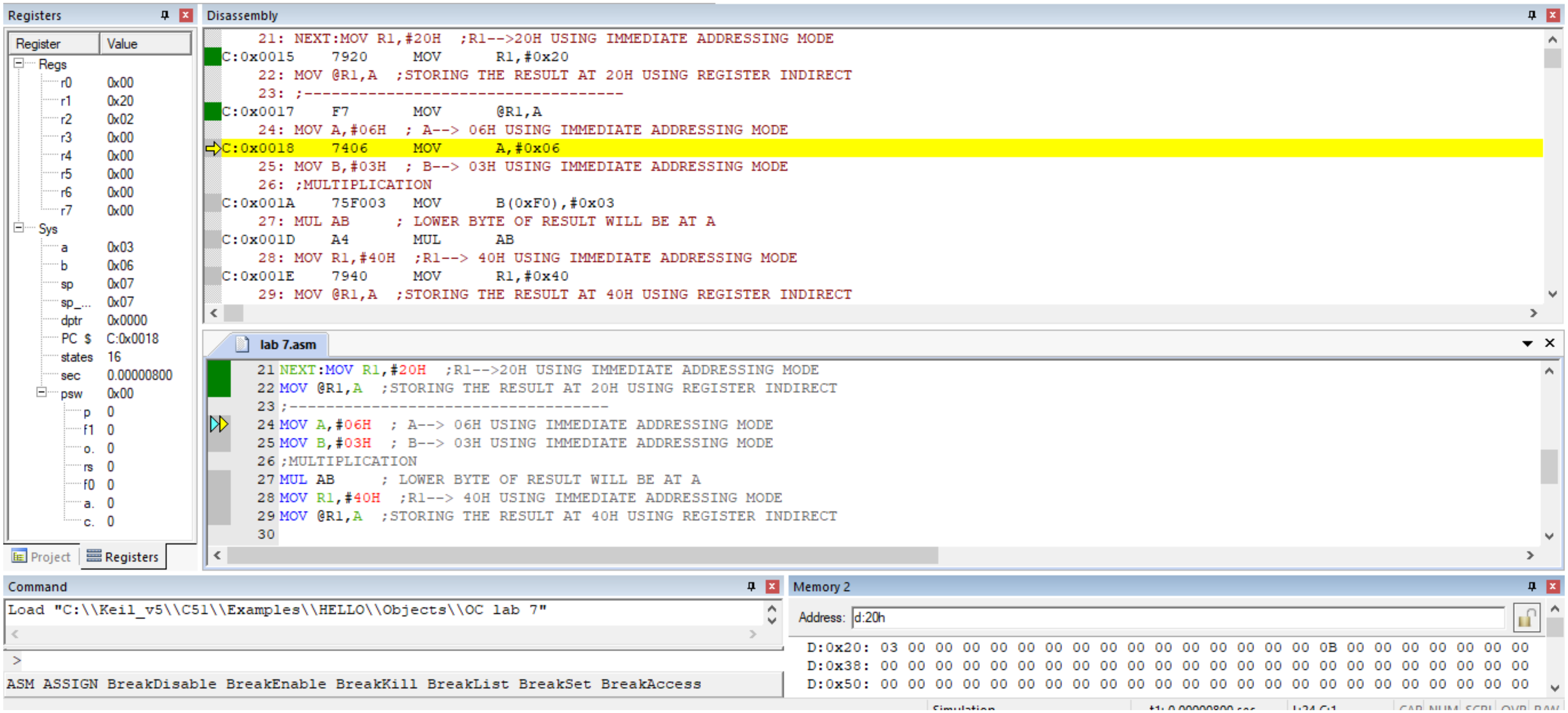
```
MOV R1,#30H   ;R1--> 30H USING IMMEDIATE ADDRESSING MODE
MOV @R1,A     ;STORING THE RESULT AT 30H USING REGISTER INDIRECT
;-----------------------------------
MOV A,#05H  ;A--> 05H USING IMMEDIATE ADDRESSING MODE
MOV R2,#02H  ; B--> 02H USING IMMEDIATE ADDRESSING MODE

;SUBTRACTION
CLR C   ;CLEARING CARRY FLAG
SUBB A,R2  ;A-->A-R2-CARRY
JNC NEXT
```

```
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\OC lab 7"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Address: d:30h

```
D:0x30: 0B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x48: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

After subtraction:



Registers

```
Register        Value
⊟ Regs
    r0          0x00
    r1          0x20
    r2          0x02
    r3          0x00
    r4          0x00
    r5          0x00
    r6          0x00
    r7          0x00
⊟ Sys
    a           0x03
    b           0x06
    sp          0x07
    sp_...      0x07
    dptr        0x0000
    PC  $       C:0x0018
    states      16
    sec         0.00000800
  ⊟ psw         0x00
      p         0
      f1        0
      o.        0
      rs        0
      f0        0
      a.        0
      c.        0
```

Disassembly

```
    21: NEXT:MOV R1,#20H   ;R1-->20H USING IMMEDIATE ADDRESSING MODE
C:0x0015     7920      MOV      R1,#0x20
    22: MOV @R1,A  ;STORING THE RESULT AT 20H USING REGISTER INDIRECT
    23: ;-----------------------------------
C:0x0017     F7        MOV      @R1,A
    24: MOV A,#06H  ; A--> 06H USING IMMEDIATE ADDRESSING MODE
C:0x0018     7406      MOV      A,#0x06
    25: MOV B,#03H  ; B--> 03H USING IMMEDIATE ADDRESSING MODE
    26: ;MULTIPLICATION
C:0x001A     75F003    MOV      B(0xF0),#0x03
    27: MUL AB      ; LOWER BYTE OF RESULT WILL BE AT A
C:0x001D     A4        MUL      AB
    28: MOV R1,#40H  ;R1--> 40H USING IMMEDIATE ADDRESSING MODE
C:0x001E     7940      MOV      R1,#0x40
    29: MOV @R1,A  ;STORING THE RESULT AT 40H USING REGISTER INDIRECT
```

lab 7.asm

```
NEXT:MOV R1,#20H   ;R1-->20H USING IMMEDIATE ADDRESSING MODE
MOV @R1,A  ;STORING THE RESULT AT 20H USING REGISTER INDIRECT
;-----------------------------------
MOV A,#06H  ; A--> 06H USING IMMEDIATE ADDRESSING MODE
MOV B,#03H  ; B--> 03H USING IMMEDIATE ADDRESSING MODE
;MULTIPLICATION
MUL AB      ; LOWER BYTE OF RESULT WILL BE AT A
MOV R1,#40H  ;R1--> 40H USING IMMEDIATE ADDRESSING MODE
MOV @R1,A  ;STORING THE RESULT AT 40H USING REGISTER INDIRECT
```

Command

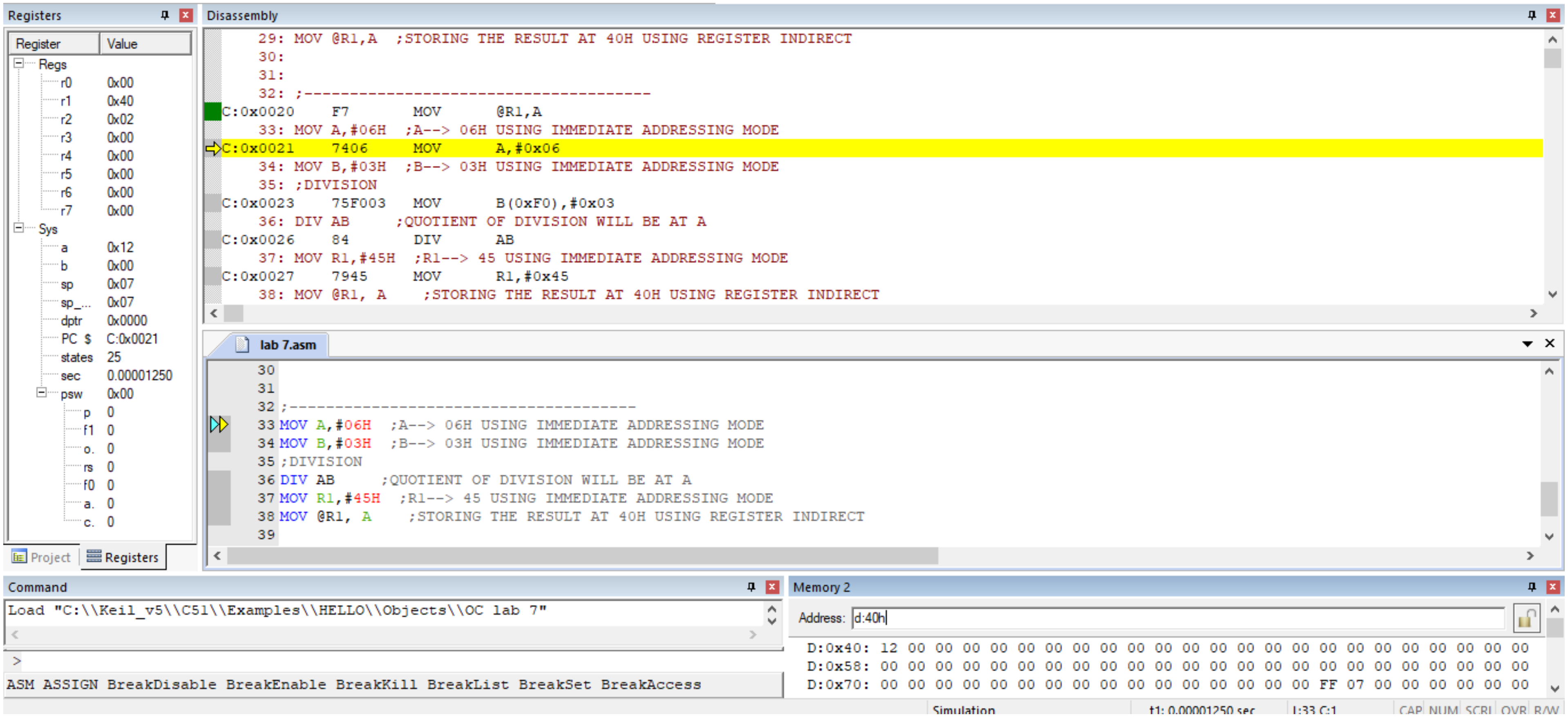
```
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\OC lab 7"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Memory 2

Address: d:20h

```
D:0x20: 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B 00 00 00 00 00 00 00
D:0x38: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

After multiplication:



Register	Value
Regs	
r0	0x00
r1	0x40
r2	0x02
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x12
b	0x00
sp	0x07
sp_...	0x07
dptr	0x0000
PC $	C:0x0021
states	25
sec	0.00001250
psw	0x00
p	0
f1	0
o.	0
rs	0
f0	0
a.	0
c.	0

```
    29: MOV @R1,A  ;STORING THE RESULT AT 40H USING REGISTER INDIRECT
    30:
    31:
    32: ;---------------------------------------
C:0x0020    F7       MOV      @R1,A
    33: MOV A,#06H  ;A--> 06H USING IMMEDIATE ADDRESSING MODE
C:0x0021    7406     MOV      A,#0x06
    34: MOV B,#03H  ;B--> 03H USING IMMEDIATE ADDRESSING MODE
    35: ;DIVISION
C:0x0023    75F003   MOV      B(0xF0),#0x03
    36: DIV AB     ;QUOTIENT OF DIVISION WILL BE AT A
C:0x0026    84       DIV      AB
    37: MOV R1,#45H  ;R1--> 45 USING IMMEDIATE ADDRESSING MODE
C:0x0027    7945     MOV      R1,#0x45
    38: MOV @R1, A    ;STORING THE RESULT AT 40H USING REGISTER INDIRECT
```

```


;---------------------------------------
MOV A,#06H  ;A--> 06H USING IMMEDIATE ADDRESSING MODE
MOV B,#03H  ;B--> 03H USING IMMEDIATE ADDRESSING MODE
;DIVISION
DIV AB     ;QUOTIENT OF DIVISION WILL BE AT A
MOV R1,#45H  ;R1--> 45 USING IMMEDIATE ADDRESSING MODE
MOV @R1, A    ;STORING THE RESULT AT 40H USING REGISTER INDIRECT

```

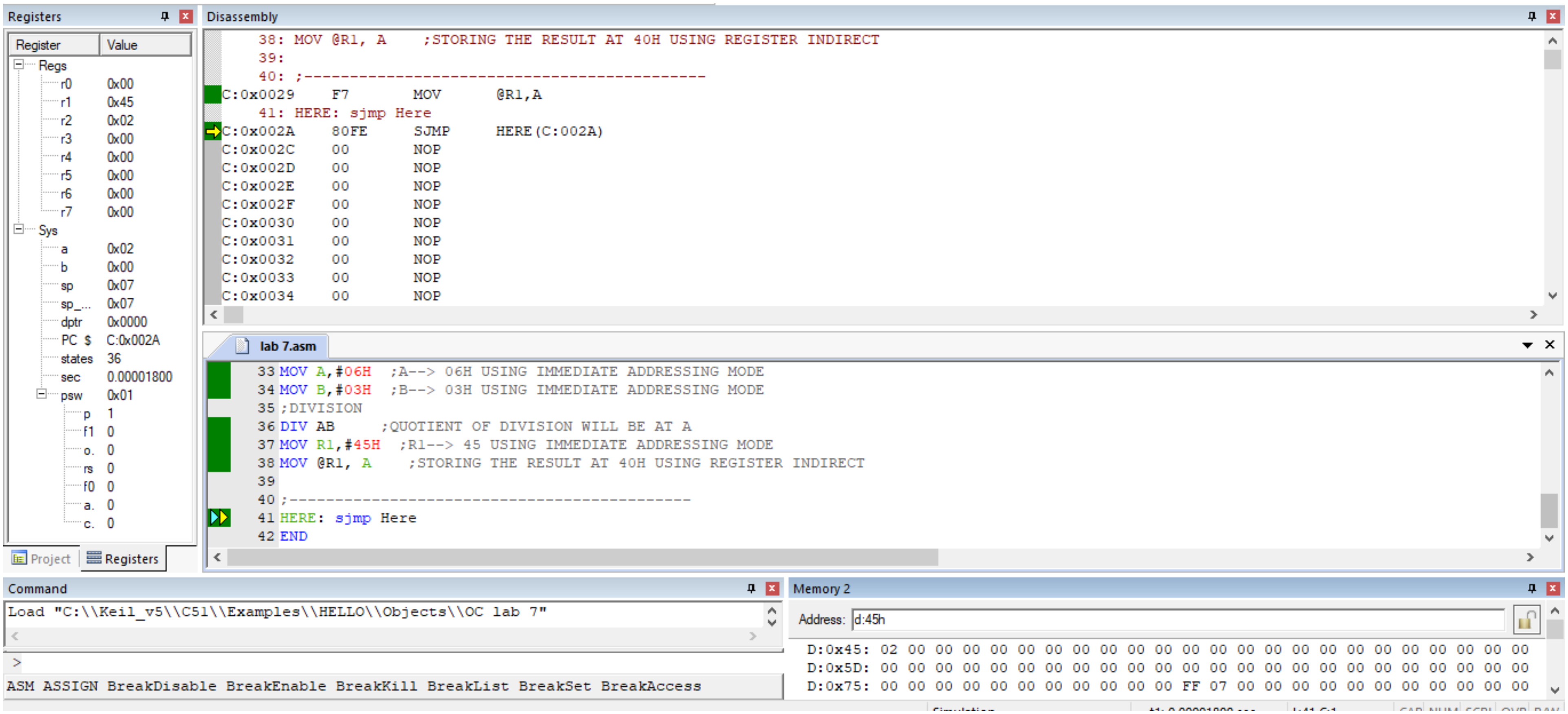
```
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\OC lab 7"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Address: d:40h

```
D:0x40: 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x58: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 07 00 00 00 00 00 00
```

Simulation t1: 0.00001250 sec L:33 C:1 CAP NUM SCRL OVR R/W

After division:



```
Register    Value
Regs
  r0        0x00
  r1        0x45
  r2        0x02
  r3        0x00
  r4        0x00
  r5        0x00
  r6        0x00
  r7        0x00
Sys
  a         0x02
  b         0x00
  sp        0x07
  sp_...    0x07
  dptr      0x0000
  PC $      C:0x002A
  states    36
  sec       0.00001800
  psw       0x01
    p       1
    f1      0
    o.      0
    rs      0
    f0      0
    a.      0
    c.      0
```

```
    38: MOV @R1, A    ;STORING THE RESULT AT 40H USING REGISTER INDIRECT
    39:
    40: ;-------------------------------------------
C:0x0029    F7       MOV      @R1,A
    41: HERE: sjmp Here
C:0x002A    80FE     SJMP     HERE(C:002A)
C:0x002C    00       NOP
C:0x002D    00       NOP
C:0x002E    00       NOP
C:0x002F    00       NOP
C:0x0030    00       NOP
C:0x0031    00       NOP
C:0x0032    00       NOP
C:0x0033    00       NOP
C:0x0034    00       NOP
```

```
MOV A,#06H   ;A--> 06H USING IMMEDIATE ADDRESSING MODE
MOV B,#03H   ;B--> 03H USING IMMEDIATE ADDRESSING MODE
;DIVISION
DIV AB      ;QUOTIENT OF DIVISION WILL BE AT A
MOV R1,#45H  ;R1--> 45 USING IMMEDIATE ADDRESSING MODE
MOV @R1, A    ;STORING THE RESULT AT 40H USING REGISTER INDIRECT

;-------------------------------------------
HERE: sjmp Here
END
```

```
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\OC lab 7"

>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

```
Address: d:45h

D:0x45: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x5D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x75: 00 00 00 00 00 00 00 00 00 00 00 FF 07 00 00 00 00 00 00 00 00 00 00 00
```