Experiment No :- 1

Aim:- Assignment exploiting the various addressing modes of 8051 for accessing internal as well as external memory.

Part 1:-

Addition of internal RAM with carry:

Theory :-

Internal RAM in 8051 microcontroller: There are 128 bytes of RAM inside the 8051 are assigned addresses 00 to 7FH. As we will see in chapter They came be accessed directly as ~~memory~~ locations. There 128 bytes are divided intro three different groups as follows!

1) A total of 32 bytes from locations 00 to 1FH are set aside for register banks & the stack.

2) A total of 16 bytes from locations 30H to 7FH are wed & write storage. or what is normally called a scratch pad. These 80 locations of RAM are widely wed for the purpose of storing data & parameters by 8051 programmers

Instructions:

1) JNC :- ~~JMP~~ Jump if no carry jumps if (Y=0) In this instraction carry flag bit in flag PSW Register is used to make decision wheales to jump.

- In execution "JNC label" processor looks at carry flag to see if it raised (CY=1) If is it not. CPU starts fetching & executing instruction from address of label.

- If CY=1 it will not jump but will execute next instruction below JNC.

2) DJNZ :- (Decrement Register & Jump if not zero)

- DJNZ decrements the value of Register by 1. If the initial value of Register is 0, decrementing the value will cause it to reset to 255. If the new value of register is not ~~a~~ the program will branch to the address indicated by relative address.

3) INC : (Increment)

- INC increments the value of Register by 1
- If the initial value of Register If 255, incrementing the value will cause it to reset to 0.
- The carry flag is not set when the value rolls over from 255 to 0.

4) ADDC :-

- The ADDC instruction adds a byte value & the value of the carry flag to the accumulator
- The results of the addition are stored back in the accumulator.

For eg:- ADDC A, Rr - (Add containts of Accumulator & Rr register & carry flag put the result in accumulator

• PSW (program status word) Register:-

CY	AC	F0	RS1	RS0	OV	–	P

1) CY PSW.7 - Carry Flag.

2) AC PSW 6 - Auxillary carry flag.

3) F0 PSW 5 - Available to the user for general purpose.

4) RS1 PSW.4 - Register Bank Selector Bit 1.
5) RS0 PSW.3 - Register Bank selector Bit 0
6) OV PSW.2 - overflow Flag.
7) - PSW.1 - user difinable bit.
8) P PSW.0 - parity Flag.

- Internal RAM addition:

- Algorithm:

1) load value 40H in R0
2) load counter.
3) Clear Accumulator.
4) clear R7.
5) Add the byte pointer to A by R0
6) If cy=0 don't accumulator carry.
7) Increment R7.
8) Increment R0 (Pointer).
9) Repeat until R0 is zero

- program:

```
        MOV R0, #40H
        MOV R2, #5
        CLR A
        MOV R7, A
Again:  ADD A, @R0
        JNC Next.
        INC R7
Next:   INC R0
        DJNZ R7, Again
        END.
```

part 2 :

a) External RAM addition:

- program:

```
          ORG 0000H
          MOV  DPTR, #200H.
          MOV  a, #00H
          Mov  R7, #00H
          MOV  r6, #08H.
Back :    MOVX A, @dptr.
          Add  a, r7
          Mov  r7, a
          inc  dptr
          djnz r6, Back
           END.
```

b) External RAM Addition with carry.

- program:

```
          ORG 0000H.
          MOV DPTR, #200H
          MOV  A , #00H
          MOV  R2, #00H
          Mov  R7, #00H
          MOV  R6, #08H
Back:     MOVX A , @DPTR.
          ADD  A , R7
          MOV  R7 , A
          JNC  Next.
          INC  R2
Next :    Inc  DPTR
          DJNZ R6, BACK
           END.
```

3) F0 PSW >

part3 :

Internal to External Block transfer:

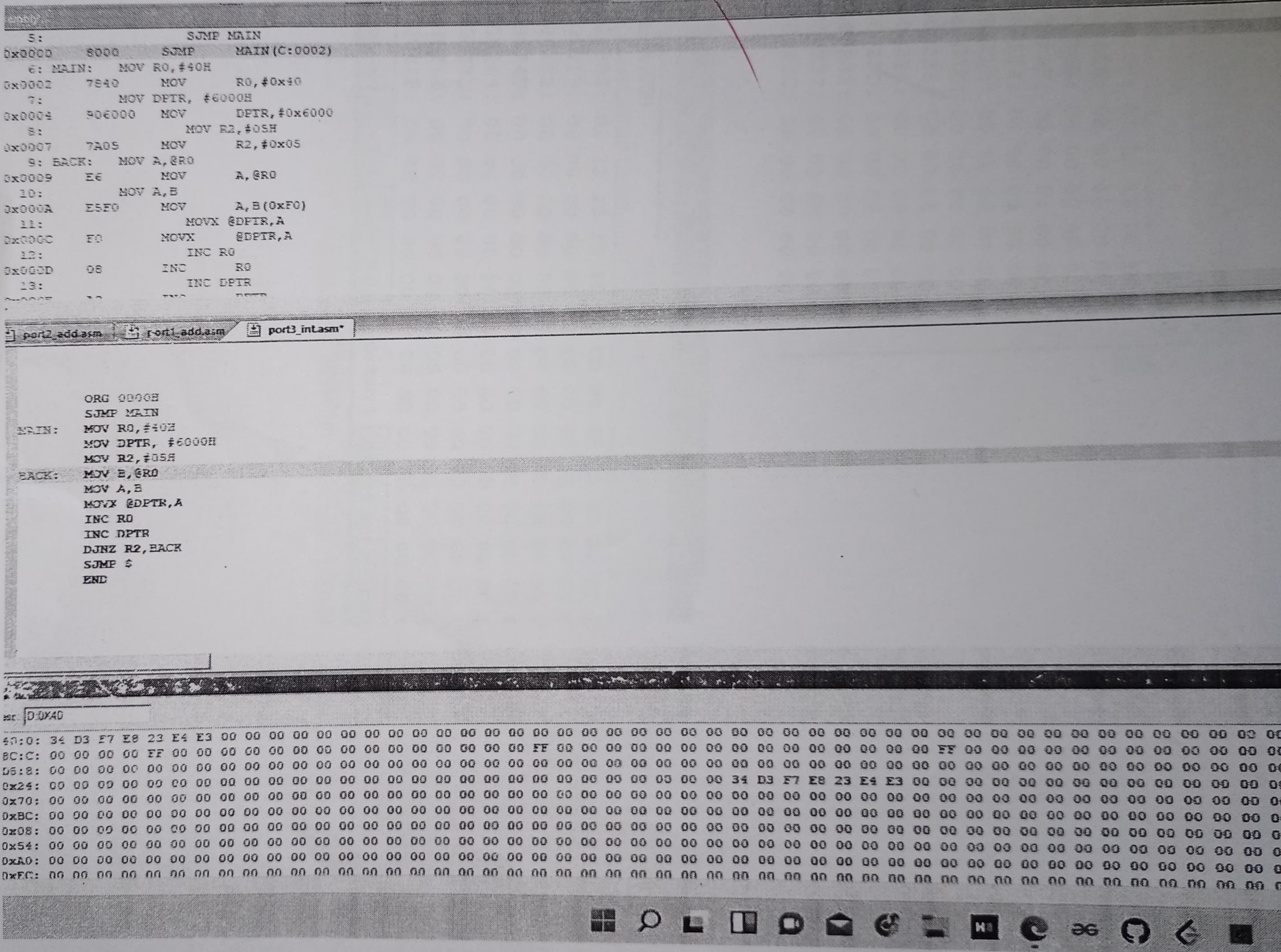
program:

```
        ORG  0000H.
        SJmp MAIN
MAIN : MOV RO , # 40H
        MOV DPTR , # 6000H
        MOV  R2 , # 05H
BACK  : MOV , @RO
        MOV  A , B.
        MOVX  @ DPTR , A
        INC   RO
        INC   DPTR
        DJNZ  R2, BACK
        SJMp  $
        END.
```

Conclusion :

In this experiment we conclude that addressing mode of 8051 for accessing internal as well as external memory addition & transfer Data internal to external.

5/5/22



```
    5:                  SJMP MAIN
0x0000    8000      SJMP      MAIN(C:0002)
    6: MAIN:   MOV R0,#40H
0x0002    7840      MOV       R0,#0x40
    7:         MOV DPTR, #6000H
0x0004    906000    MOV       DPTR,#0x6000
    8:                  MOV R2,#05H
0x0007    7A05      MOV       R2,#0x05
    9: BACK:   MOV A,@R0
0x0009    E6        MOV       A,@R0
   10:         MOV A,B
0x000A    E5F0      MOV       A,B(0xF0)
   11:                  MOVX @DPTR,A
0x000C    F0        MOVX      @DPTR,A
   12:                  INC R0
0x000D    08        INC       R0
   13:                  INC DPTR
```

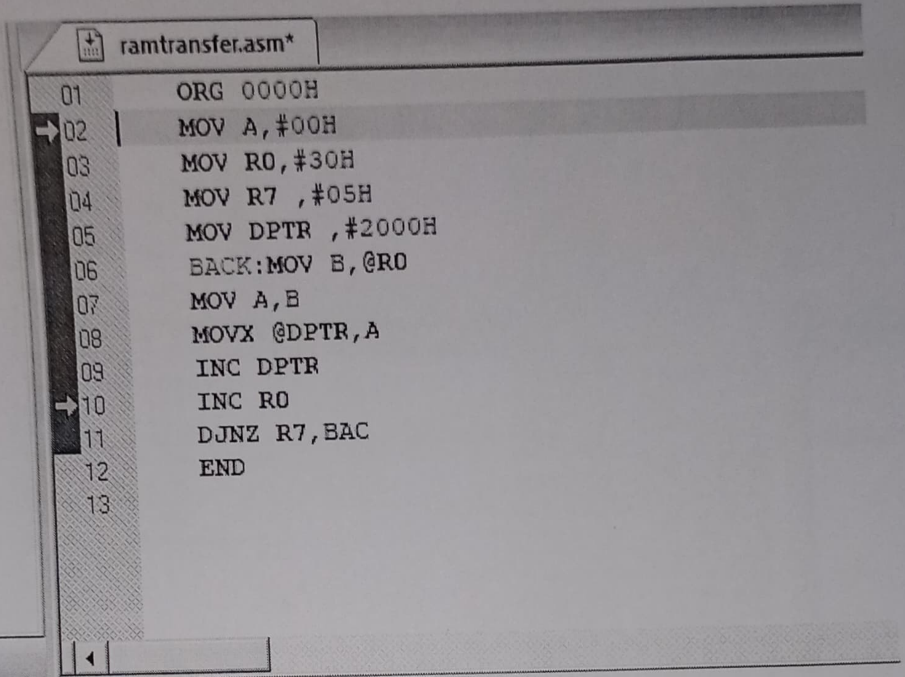
```
        ORG 0000H
        SJMP MAIN
MAIN:   MOV R0,#40H
        MOV DPTR, #6000H
        MOV R2,#05H
BACK:   MOV B,@R0
        MOV A,B
        MOVX @DPTR,A
        INC R0
        INC DPTR
        DJNZ R2,BACK
        SJMP $
        END
```

```
ss: D:0X40
40:0: 34 D3 F7 E8 23 E4 E3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
BC:C: 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 0
D6:8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0x24: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 34 D3 F7 E8 23 E4 E3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0x70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0xBC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0x08: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0x54: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0xA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0xEC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
```

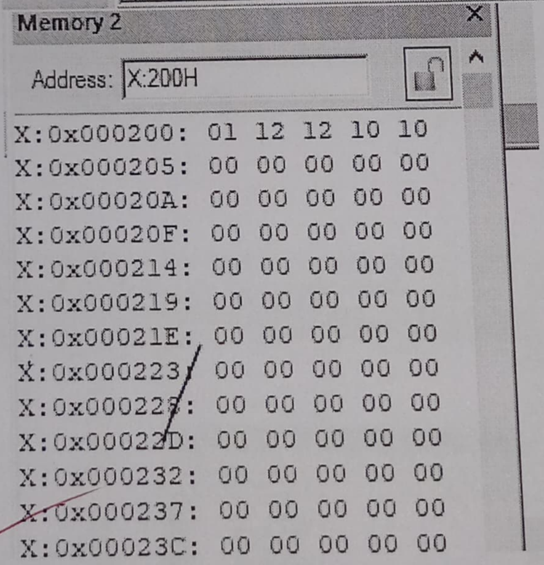
)

00

0000

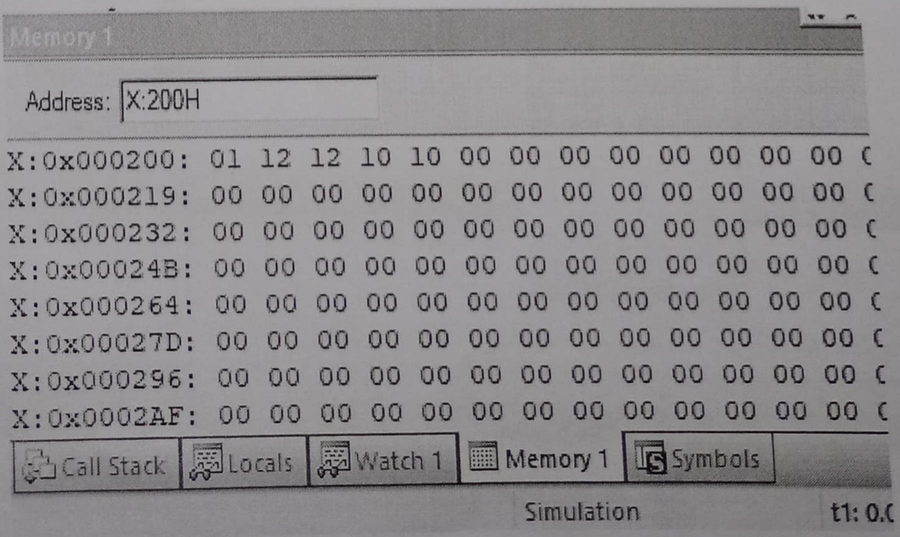


```
      ORG 0000H
      MOV A,#00H
      MOV R0,#30H
      MOV R7 ,#05H
      MOV DPTR ,#2000H
      BACK:MOV B,@R0
      MOV A,B
      MOVX @DPTR,A
      INC DPTR
      INC R0
      DJNZ R7,BAC
      END
```



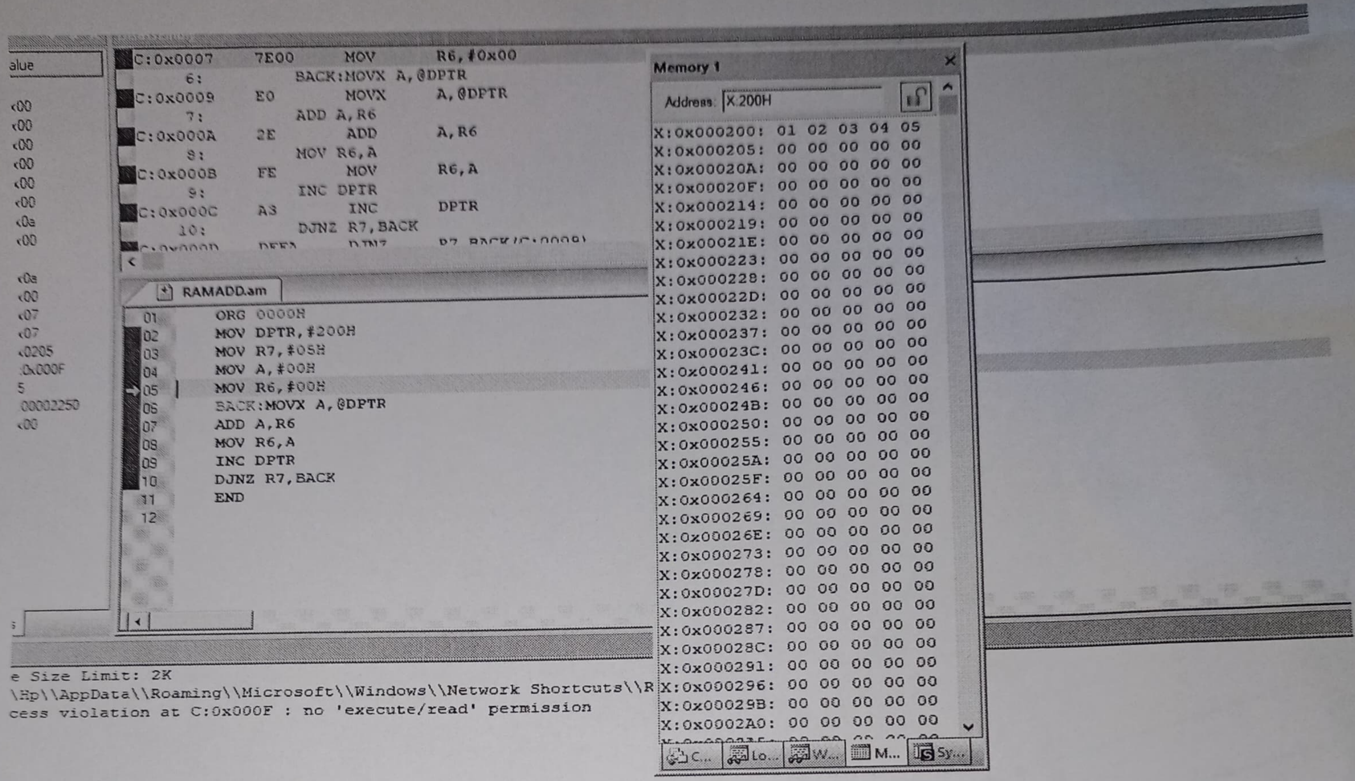
Address: X:200H

```
X:0x000200: 01 12 12 10 10
X:0x000205: 00 00 00 00 00
X:0x00020A: 00 00 00 00 00
X:0x00020F: 00 00 00 00 00
X:0x000214: 00 00 00 00 00
X:0x000219: 00 00 00 00 00
X:0x00021E: 00 00 00 00 00
X:0x000223: 00 00 00 00 00
X:0x000228: 00 00 00 00 00
X:0x00022D: 00 00 00 00 00
X:0x000232: 00 00 00 00 00
X:0x000237: 00 00 00 00 00
X:0x00023C: 00 00 00 00 00
```



Address: X:200H

```
X:0x000200: 01 12 12 10 10 00 00 00 00 00 00 00 00
X:0x000219: 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000232: 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x00024B: 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000264: 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x00027D: 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000296: 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x0002AF: 00 00 00 00 00 00 00 00 00 00 00 00 00
```



```
alue
x00
x00
x00
x00
x00
x00
x0a
x00

x0a
x00
x07
x07
x0205
0x000F
5
00002250
x00
```

```
C:0x0007    7E00      MOV       R6,#0x00
     6:          BACK:MOVX A,@DPTR
C:0x0009    E0        MOVX      A,@DPTR
     7:          ADD A,R6
C:0x000A    2E        ADD       A,R6
     8:          MOV R6,A
C:0x000B    FE        MOV       R6,A
     9:          INC DPTR
C:0x000C    A3        INC       DPTR
    10:          DJNZ R7,BACK
```

RAMADD.am

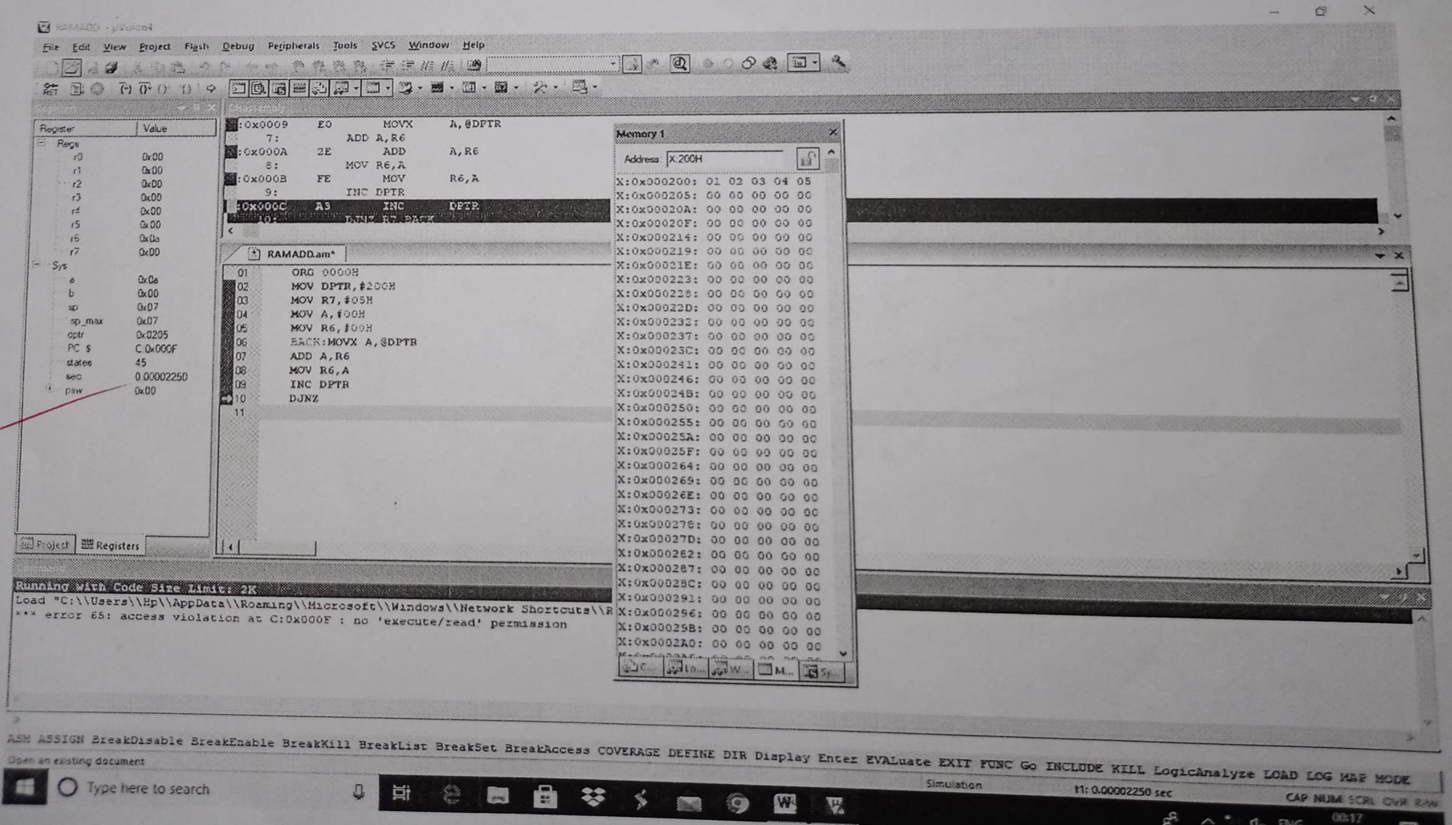
```
ORG 0000H
MOV DPTR,#200H
MOV R7,#05H
MOV A,#00H
MOV R6,#00H
BACK:MOVX A,@DPTR
ADD A,R6
MOV R6,A
INC DPTR
DJNZ R7,BACK
END
```

Memory 1

Address: X:200H

```
X:0x000200: 01 02 03 04 05
X:0x000205: 00 00 00 00 00
X:0x00020A: 00 00 00 00 00
X:0x00020F: 00 00 00 00 00
X:0x000214: 00 00 00 00 00
X:0x000219: 00 00 00 00 00
X:0x00021E: 00 00 00 00 00
X:0x000223: 00 00 00 00 00
X:0x000228: 00 00 00 00 00
X:0x00022D: 00 00 00 00 00
X:0x000232: 00 00 00 00 00
X:0x000237: 00 00 00 00 00
X:0x00023C: 00 00 00 00 00
X:0x000241: 00 00 00 00 00
X:0x000246: 00 00 00 00 00
X:0x00024B: 00 00 00 00 00
X:0x000250: 00 00 00 00 00
X:0x000255: 00 00 00 00 00
X:0x00025A: 00 00 00 00 00
X:0x00025F: 00 00 00 00 00
X:0x000264: 00 00 00 00 00
X:0x000269: 00 00 00 00 00
X:0x00026E: 00 00 00 00 00
X:0x000273: 00 00 00 00 00
X:0x000278: 00 00 00 00 00
X:0x00027D: 00 00 00 00 00
X:0x000282: 00 00 00 00 00
X:0x000287: 00 00 00 00 00
X:0x00028C: 00 00 00 00 00
X:0x000291: 00 00 00 00 00
X:0x000296: 00 00 00 00 00
X:0x00029B: 00 00 00 00 00
X:0x0002A0: 00 00 00 00 00
```

```
e Size Limit: 2K
\Hp\\AppData\\Roaming\\Microsoft\\Windows\\Network Shortcuts\\R
cess violation at C:0x000F : no 'execute/read' permission
```



File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x0a
r7	0x00
Sys	
a	0x0a
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0205
PC $	C:0x000F
states	45
sec	0.00002250
psw	0x00

```
:0x0009     E0        MOVX      A,@DPTR
     7:          ADD A,R6
:0x000A     2E        ADD       A,R6
     8:          MOV R6,A
:0x000B     FE        MOV       R6,A
     9:          INC DPTR
:0x000C     A3        INC       DPTR
    10:          DJNZ R7,BACK
```

RAMADD.am*

```
ORG 0000H
MOV DPTR,#200H
MOV R7,#05H
MOV A,#00H
MOV R6,#00H
BACK:MOVX A,@DPTR
ADD A,R6
MOV R6,A
INC DPTR
DJNZ
```

Memory 1

Address: X:200H

```
X:0x000200: 01 02 03 04 05
X:0x000205: 00 00 00 00 00
X:0x00020A: 00 00 00 00 00
X:0x00020F: 00 00 00 00 00
X:0x000214: 00 00 00 00 00
X:0x000219: 00 00 00 00 00
X:0x00021E: 00 00 00 00 00
X:0x000223: 00 00 00 00 00
X:0x000228: 00 00 00 00 00
X:0x00022D: 00 00 00 00 00
X:0x000232: 00 00 00 00 00
X:0x000237: 00 00 00 00 00
X:0x00023C: 00 00 00 00 00
X:0x000241: 00 00 00 00 00
X:0x000246: 00 00 00 00 00
X:0x00024B: 00 00 00 00 00
X:0x000250: 00 00 00 00 00
X:0x000255: 00 00 00 00 00
X:0x00025A: 00 00 00 00 00
X:0x00025F: 00 00 00 00 00
X:0x000264: 00 00 00 00 00
X:0x000269: 00 00 00 00 00
X:0x00026E: 00 00 00 00 00
X:0x000273: 00 00 00 00 00
X:0x000278: 00 00 00 00 00
X:0x00027D: 00 00 00 00 00
X:0x000282: 00 00 00 00 00
X:0x000287: 00 00 00 00 00
X:0x00028C: 00 00 00 00 00
X:0x000291: 00 00 00 00 00
X:0x000296: 00 00 00 00 00
X:0x00029B: 00 00 00 00 00
X:0x0002A0: 00 00 00 00 00
```

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Hp\\AppData\\Roaming\\Microsoft\\Windows\\Network Shortcuts\\R
*** error 65: access violation at C:0x000F : no 'execute/read' permission
```

```
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
```