Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 19-02-2021

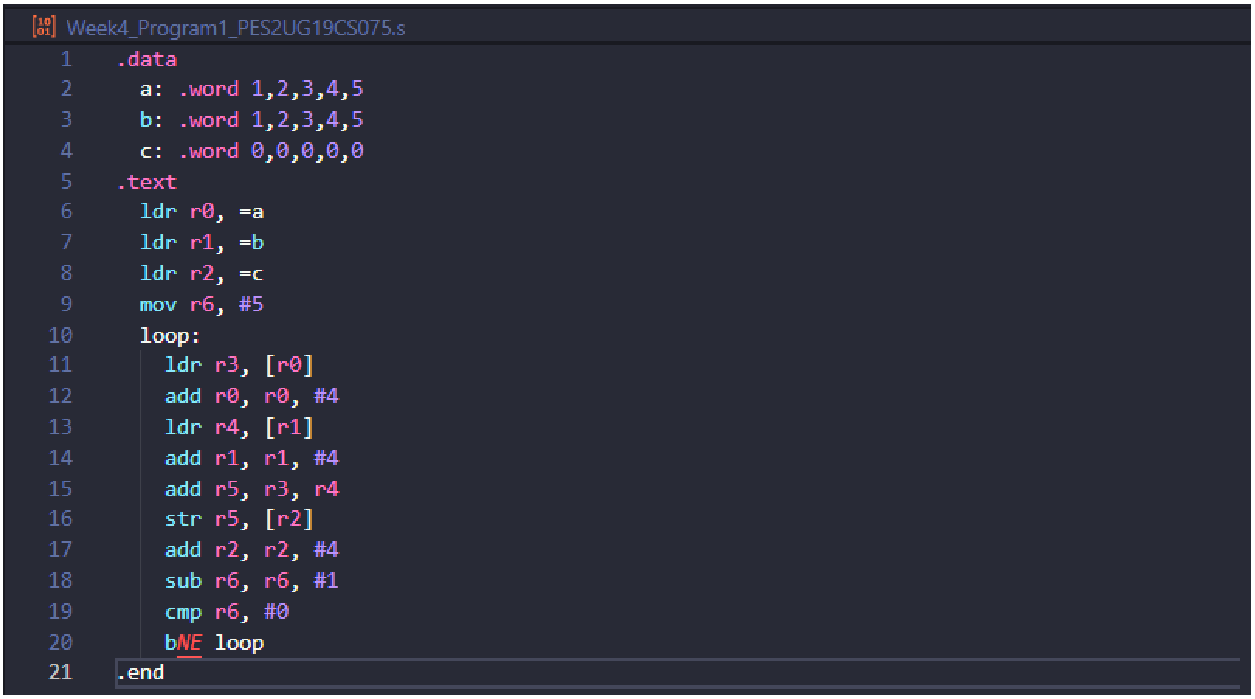
Name: Atul Anurag	SRN: PES2UG19CS075	Section: B

Week#4

Program Number: 1

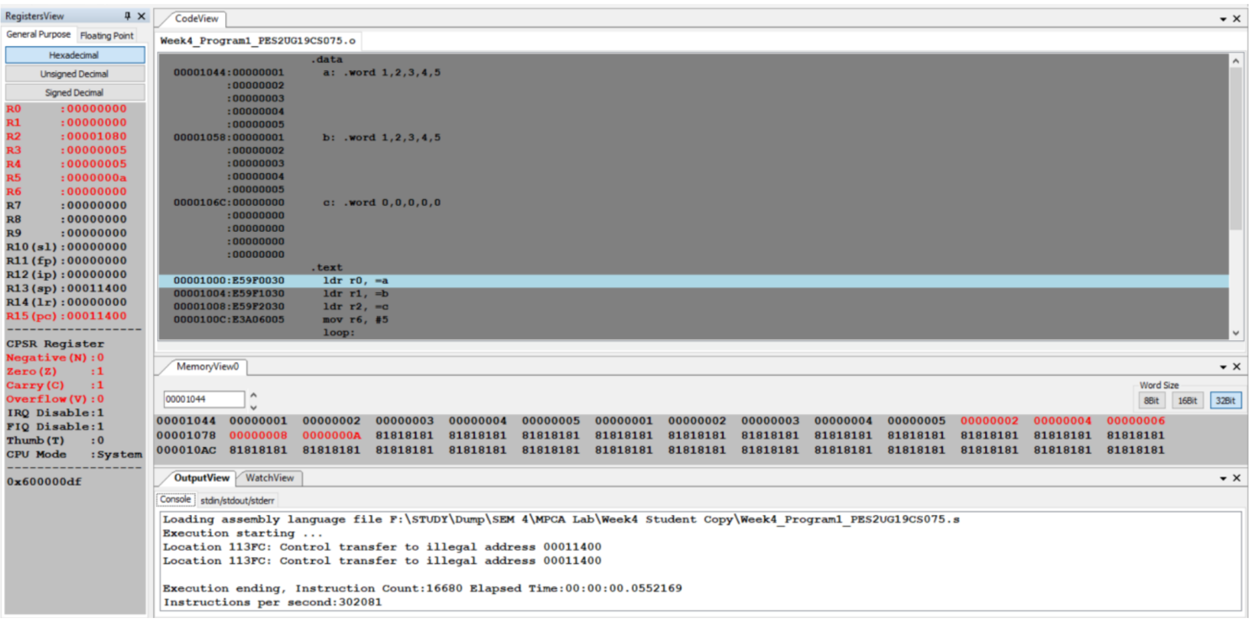
Write an ALP to implement C[k] =a[i]+b[j]

I. ARM Assembly Code (1).



```
.data
  a: .word 1,2,3,4,5
  b: .word 1,2,3,4,5
  c: .word 0,0,0,0,0
.text
  ldr r0, =a
  ldr r1, =b
  ldr r2, =c
  mov r6, #5
  loop:
    ldr r3, [r0]
    add r0, r0, #4
    ldr r4, [r1]
    add r1, r1, #4
    add r5, r3, r4
    str r5, [r2]
    add r2, r2, #4
    sub r6, r6, #1
    cmp r6, #0
    bNE loop
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0        :00000000
R1        :00000000
R2        :00001080
R3        :00000005
R4        :00000005
R5        :0000000a
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):00011400
-----------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
-----------------
0x600000df
```

```
                          .data
  00001044:00000001         a: .word 1,2,3,4,5
          :00000002
          :00000003
          :00000004
          :00000005
  00001058:00000001         b: .word 1,2,3,4,5
          :00000002
          :00000003
          :00000004
          :00000005
  0000106C:00000000         c: .word 0,0,0,0,0
          :00000000
          :00000000
          :00000000
          :00000000
                          .text
  00001000:E59F0030         ldr r0, =a
  00001004:E59F1030         ldr r1, =b
  00001008:E59F2030         ldr r2, =c
  0000100C:E3A06005         mov r6, #5
                            loop:
```

```
00001044
00001044 00000001 00000002 00000003 00000004 00000005 00000001 00000002 00000003 00000004 00000005 00000002 00000004 00000006
00001078 00000008 0000000A 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program1_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16680 Elapsed Time:00:00:00.0552169
Instructions per second:302081
```

III. Output Table for the program(1)

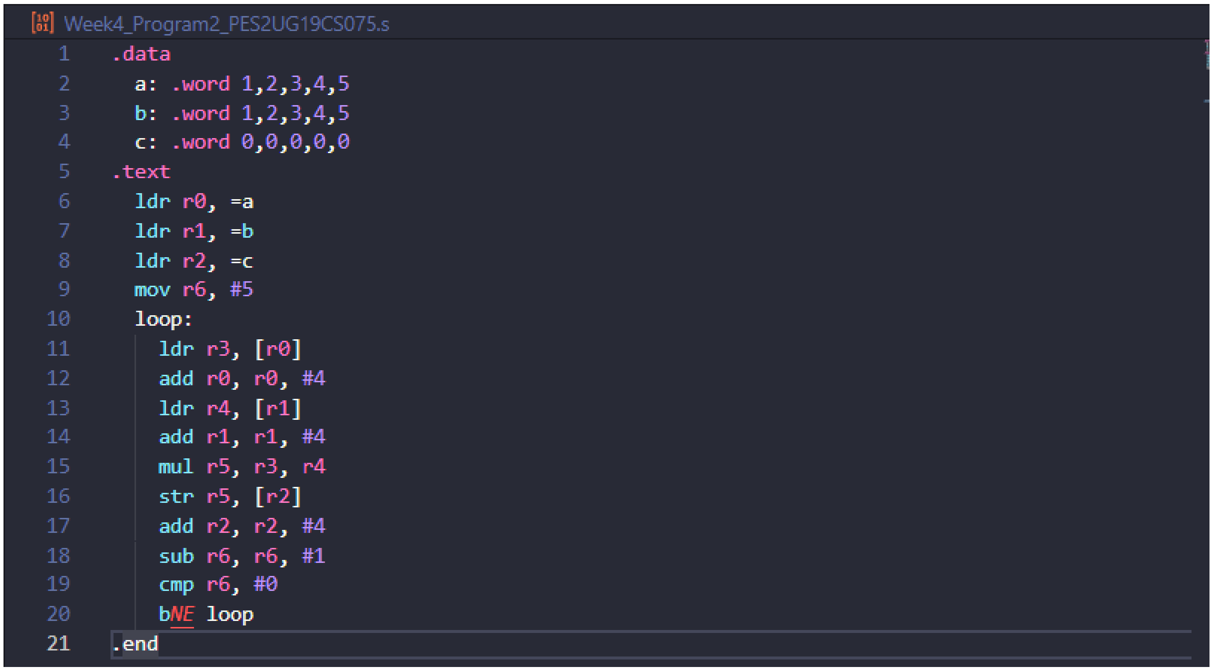
<table>
<tr><th colspan="2">a: .word 1, 2, 3, 4, 5
b: .word 1, 2, 3, 4, 5
c: .word 0,0,0,0,0</th></tr>
<tr><th colspan="2">After Execution The content of array C is</th></tr>
<tr><td>2</td><td>00000002</td></tr>
<tr><td>4</td><td>00000004</td></tr>
<tr><td>6</td><td>00000006</td></tr>
<tr><td>8</td><td>00000008</td></tr>
<tr><td>10</td><td>0000000A</td></tr>
</table>

Week#4

Program Number: 2

Write an ALP to implement c[k] = a[i] * b[j]

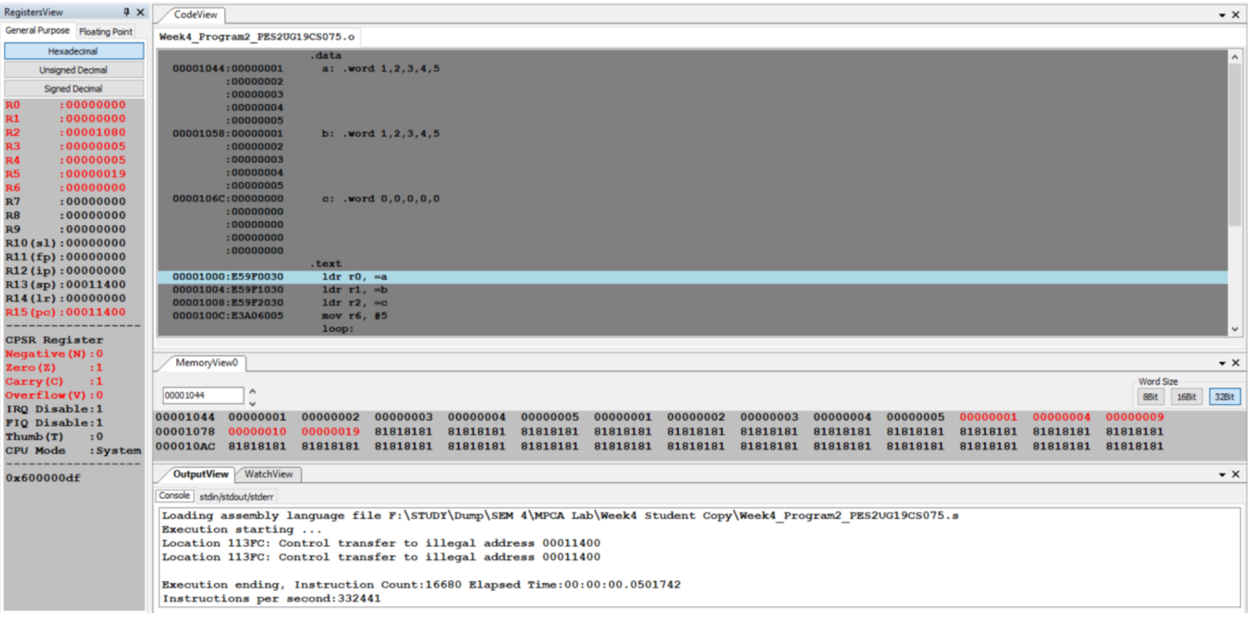
I. ARM Assembly Code (1).



Week4_Program2_PES2UG19CS075.s

```
.data
  a: .word 1,2,3,4,5
  b: .word 1,2,3,4,5
  c: .word 0,0,0,0,0
.text
  ldr r0, =a
  ldr r1, =b
  ldr r2, =c
  mov r6, #5
  loop:
    ldr r3, [r0]
    add r0, r0, #4
    ldr r4, [r1]
    add r1, r1, #4
    mul r5, r3, r4
    str r5, [r2]
    add r2, r2, #4
    sub r6, r6, #1
    cmp r6, #0
    bNE loop
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0       :00000000
R1       :00000000
R2       :00001080
R3       :00000005
R4       :00000005
R5       :00000019
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00000000
R15(pc)  :00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
Week4_Program2_PES2UG19CS075.o
                            .data
00001044:00000001             a: .word 1,2,3,4,5
        :00000002
        :00000003
        :00000004
        :00000005
00001058:00000001             b: .word 1,2,3,4,5
        :00000002
        :00000003
        :00000004
        :00000005
0000106C:00000000             c: .word 0,0,0,0,0
        :00000000
        :00000000
        :00000000
        :00000000
                            .text
00001000:E59F0030             ldr r0, =a
00001004:E59F1030             ldr r1, =b
00001008:E59F2030             ldr r2, =c
0000100C:E3A06005             mov r6, #5
                              loop:
```

```
00001044
00001044 00000001 00000002 00000003 00000004 00000005 00000001 00000002 00000003 00000004 00000005 00000001 00000004 00000009
00001078 00000010 00000019 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program2_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16680 Elapsed Time:00:00:00.0501742
Instructions per second:332441
```

III. Output Table for the program(1)

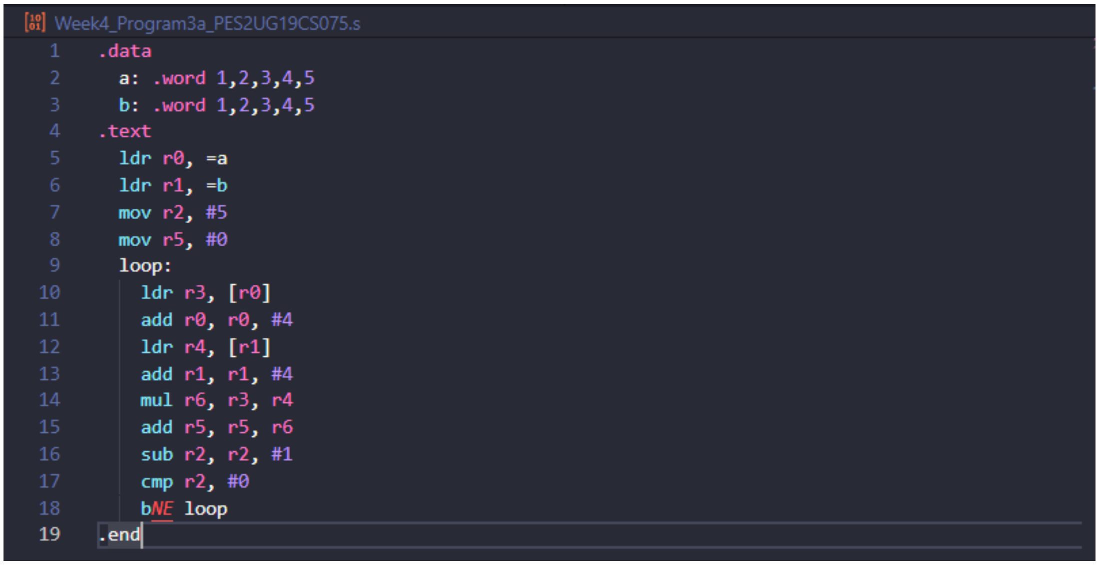
a: .word 1, 2, 3, 4, 5 b: .word 1, 2, 3, 4, 5 c: .word 0,0,0,0,0	
After Execution The content of array C is	
1	**00000001**
4	**00000004**
9	**00000009**
16	**00000010**
25	**00000019**

Week#4

Program Number: 3

a. Write an ALP to perform Convolution using MUL instruction (Addition of multiplication of respective numbers of loc A and loc B)

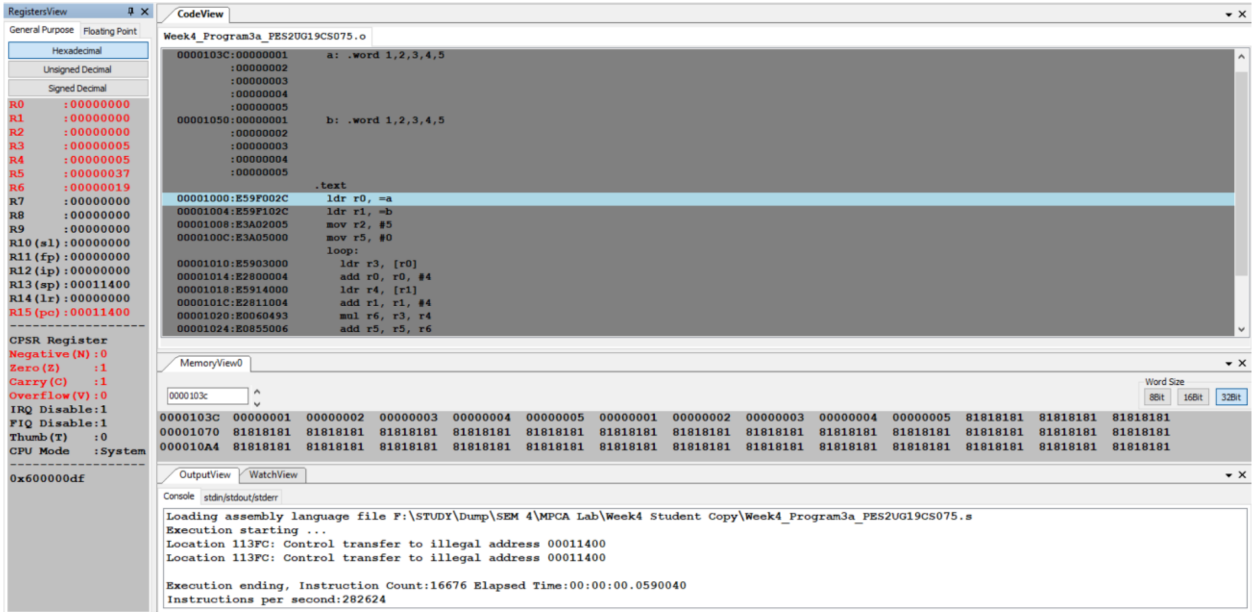
I. ARM Assembly Code (1).



Week4_Program3a_PES2UG19CS075.s

```
.data
  a: .word 1,2,3,4,5
  b: .word 1,2,3,4,5
.text
  ldr r0, =a
  ldr r1, =b
  mov r2, #5
  mov r5, #0
  loop:
    ldr r3, [r0]
    add r0, r0, #4
    ldr r4, [r1]
    add r1, r1, #4
    mul r6, r3, r4
    add r5, r5, r6
    sub r2, r2, #1
    cmp r2, #0
    bNE loop
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0       :00000000
R1       :00000000
R2       :00000000
R3       :00000005
R4       :00000005
R5       :00000037
R6       :00000019
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00000000
R15(pc)  :00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

Week4_Program3a_PES2UG19CS075.o

```
0000103C:00000001      a: .word 1,2,3,4,5
        :00000002
        :00000003
        :00000004
        :00000005
00001050:00000001      b: .word 1,2,3,4,5
        :00000002
        :00000003
        :00000004
        :00000005
                     .text
00001000:E59F002C      ldr r0, =a
00001004:E59F102C      ldr r1, =b
00001008:E3A02005      mov r2, #5
0000100C:E3A05000      mov r5, #0
                       loop:
00001010:E5903000        ldr r3, [r0]
00001014:E2800004        add r0, r0, #4
00001018:E5914000        ldr r4, [r1]
0000101C:E2811004        add r1, r1, #4
00001020:E0060493        mul r6, r3, r4
00001024:E0855006        add r5, r5, r6
```

0000103c

```
0000103C  00000001  00000002  00000003  00000004  00000005  00000001  00000002  00000003  00000004  00000005  81818181  81818181  81818181
00001070  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010A4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

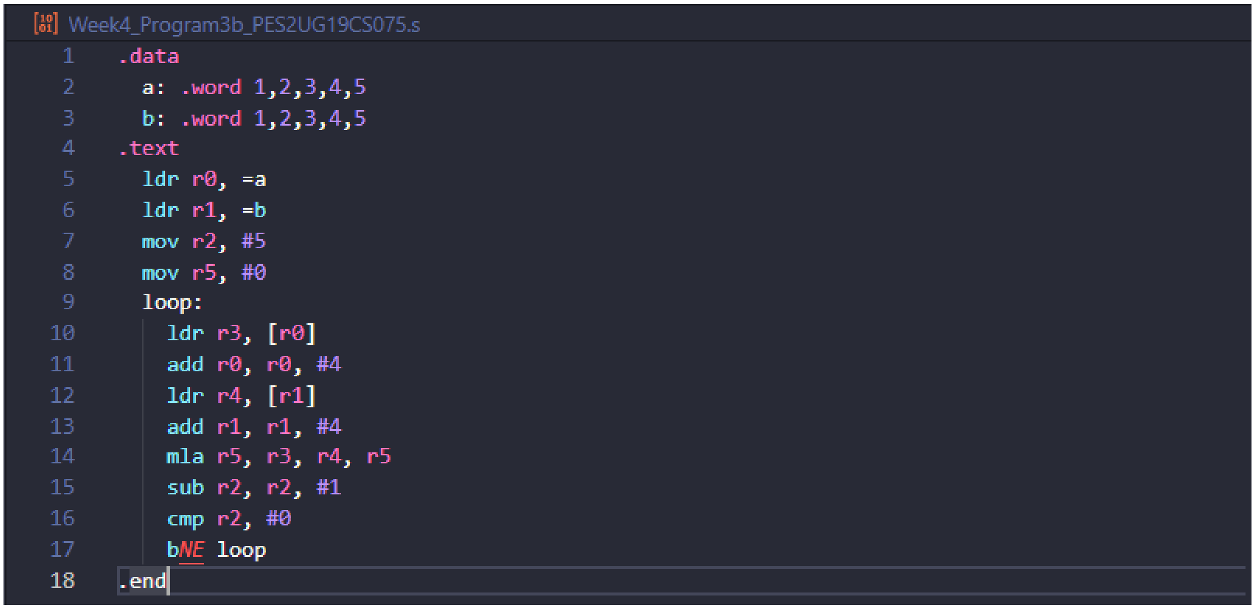
```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program3a_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16676 Elapsed Time:00:00:00.0590040
Instructions per second:282624
```

III. Output Table for the program(1)

a: .word 1, 2, 3, 4, 5 **b: .word 1, 2, 3, 4, 5**	
R5	**(1*1)+(2*2)+(3*3) +(4*4)+(5*5) =55=00000037**

b. Write an ALP to perform Convolution using MLA instruction (Addition of multiplication of respective numbers of loc A and loc B).

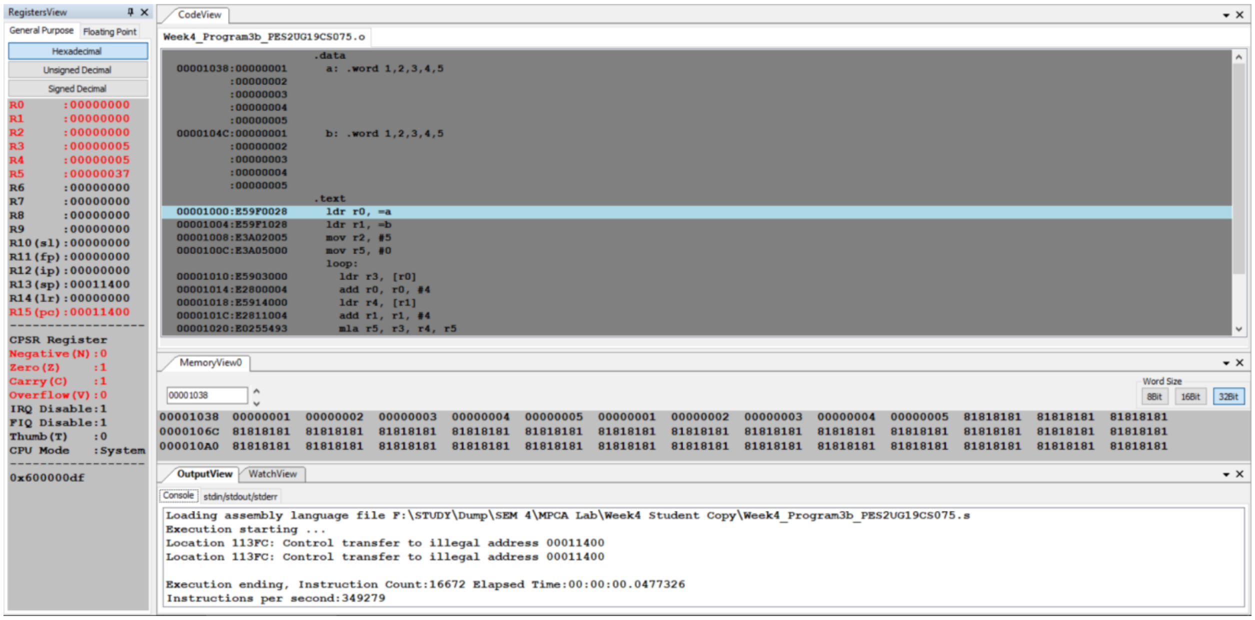
c. ARM Assembly Code (1).



Week4_Program3b_PES2UG19CS075.s

```
.data
  a: .word 1,2,3,4,5
  b: .word 1,2,3,4,5
.text
  ldr r0, =a
  ldr r1, =b
  mov r2, #5
  mov r5, #0
  loop:
    ldr r3, [r0]
    add r0, r0, #4
    ldr r4, [r1]
    add r1, r1, #4
    mla r5, r3, r4, r5
    sub r2, r2, #1
    cmp r2, #0
    bNE loop
.end
```

IV. Output Screen Shot (One Example of your choice)



```
R0        :00000000
R1        :00000000
R2        :00000000
R3        :00000005
R4        :00000005
R5        :00000037
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):00011400
-----------------
CPSR Register
Negative(N):0
Zero(Z)      :1
Carry(C)     :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)     :0
CPU Mode     :System
-----------------
0x600000df
```

Week4_Program3b_PES2UG19CS075.o

```
                          .data
  00001038:00000001         a: .word 1,2,3,4,5
          :00000002
          :00000003
          :00000004
          :00000005
  0000104C:00000001         b: .word 1,2,3,4,5
          :00000002
          :00000003
          :00000004
          :00000005
                          .text
  00001000:E59F0028         ldr r0, =a
  00001004:E59F1028         ldr r1, =b
  00001008:E3A02005         mov r2, #5
  0000100C:E3A05000         mov r5, #0
                            loop:
  00001010:E5903000           ldr r3, [r0]
  00001014:E2800004           add r0, r0, #4
  00001018:E5914000           ldr r4, [r1]
  0000101C:E2811004           add r1, r1, #4
  00001020:E0255493           mla r5, r3, r4, r5
```

```
00001038
00001038  00000001  00000002  00000003  00000004  00000005  00000001  00000002  00000003  00000004  00000005  81818181  81818181  81818181
0000106C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010A0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program3b_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16672 Elapsed Time:00:00:00.0477326
Instructions per second:349279
```

V. Output Table for the program(1)

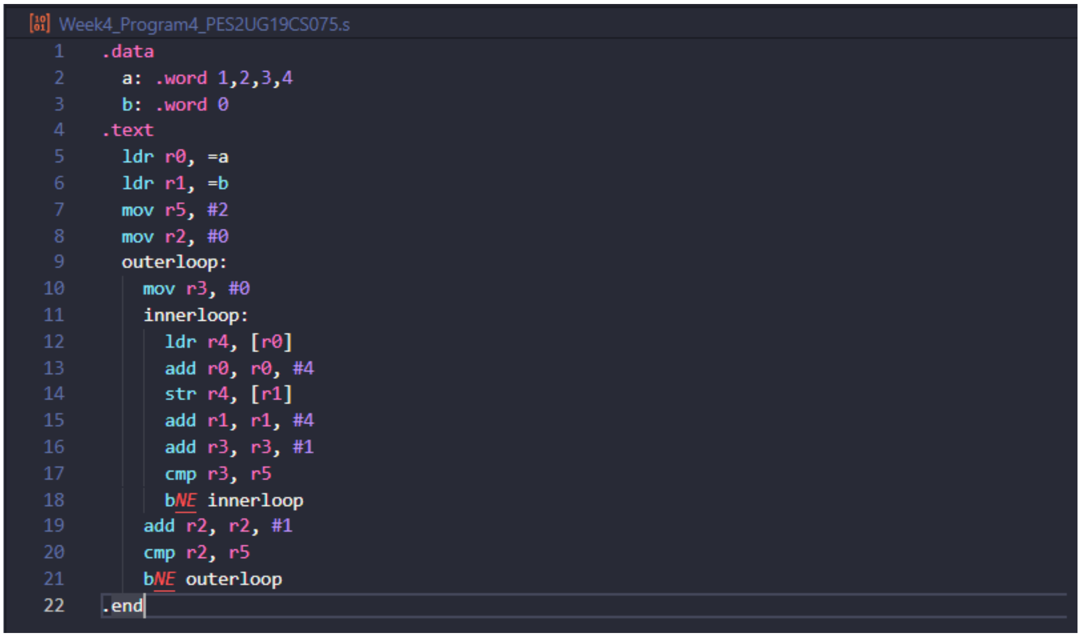
a: .word 1, 2, 3, 4, 5 b: .word 1, 2, 3, 4, 5	
R5	(1*1)+(2*2)+(3*3)+(4*4)+(5*5)=55=00000037

Week#4

Program Number: 4

Write an ALP to read from a 2D array such that B=a[i] [j]

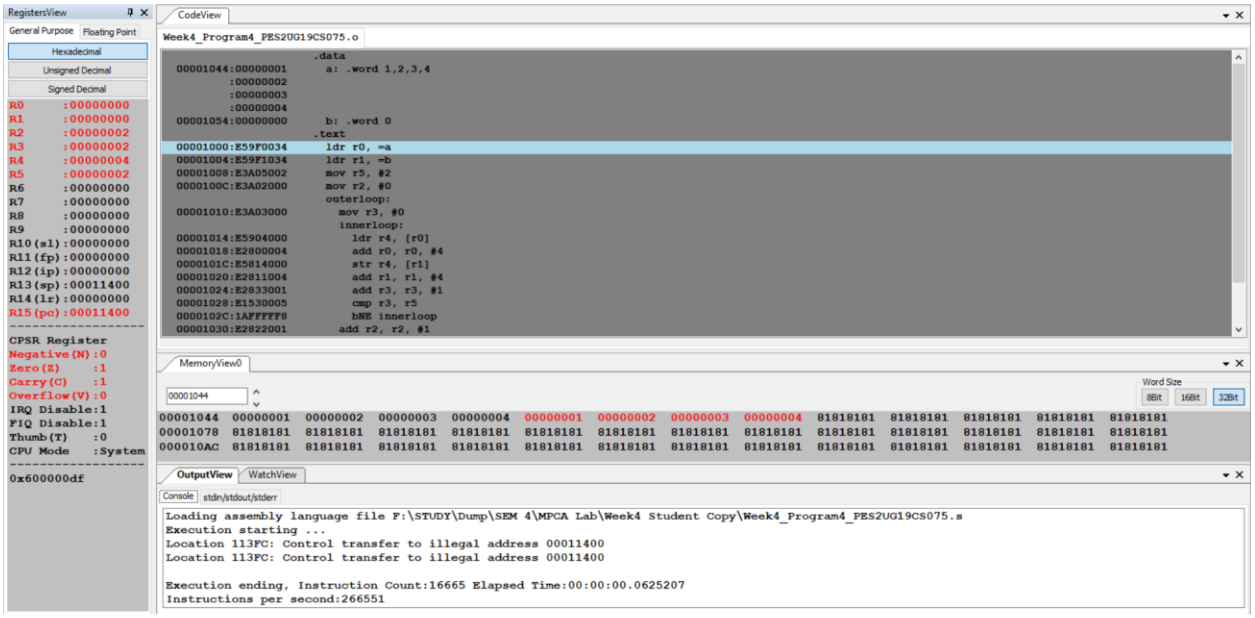
I. ARM Assembly Code (1).



Week4_Program4_PES2UG19CS075.s

```
.data
  a: .word 1,2,3,4
  b: .word 0
.text
  ldr r0, =a
  ldr r1, =b
  mov r5, #2
  mov r2, #0
  outerloop:
    mov r3, #0
    innerloop:
      ldr r4, [r0]
      add r0, r0, #4
      str r4, [r1]
      add r1, r1, #4
      add r3, r3, #1
      cmp r3, r5
      bNE innerloop
    add r2, r2, #1
    cmp r2, r5
    bNE outerloop
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0      :00000000
R1      :00000000
R2      :00000002
R3      :00000002
R4      :00000004
R5      :00000002
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00011400
R14(lr) :00000000
R15(pc) :00011400
-----------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
-----------------
0x600000df
```

Week4_Program4_PES2UG19CS075.o

```
                          .data
00001044:00000001          a: .word 1,2,3,4
        :00000002
        :00000003
        :00000004
00001054:00000000          b: .word 0
                          .text
00001000:E59F0034          ldr r0, =a
00001004:E59F1034          ldr r1, =b
00001008:E3A05002          mov r5, #2
0000100C:E3A02000          mov r2, #0
                           outerloop:
00001010:E3A03000            mov r3, #0
                             innerloop:
00001014:E5904000              ldr r4, [r0]
00001018:E2800004              add r0, r0, #4
0000101C:E5814000              str r4, [r1]
00001020:E2811004              add r1, r1, #4
00001024:E2833001              add r3, r3, #1
00001028:E1530005              cmp r3, r5
0000102C:1AFFFFF8              bNE innerloop
00001030:E2822001            add r2, r2, #1
```

00001044

```
00001044 00000001 00000002 00000003 00000004 00000001 00000002 00000003 00000004 81818181 81818181 81818181 81818181 81818181
00001078 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program4_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16665 Elapsed Time:00:00:00.0625207
Instructions per second:266551
```

III. Output Table for the program(1)

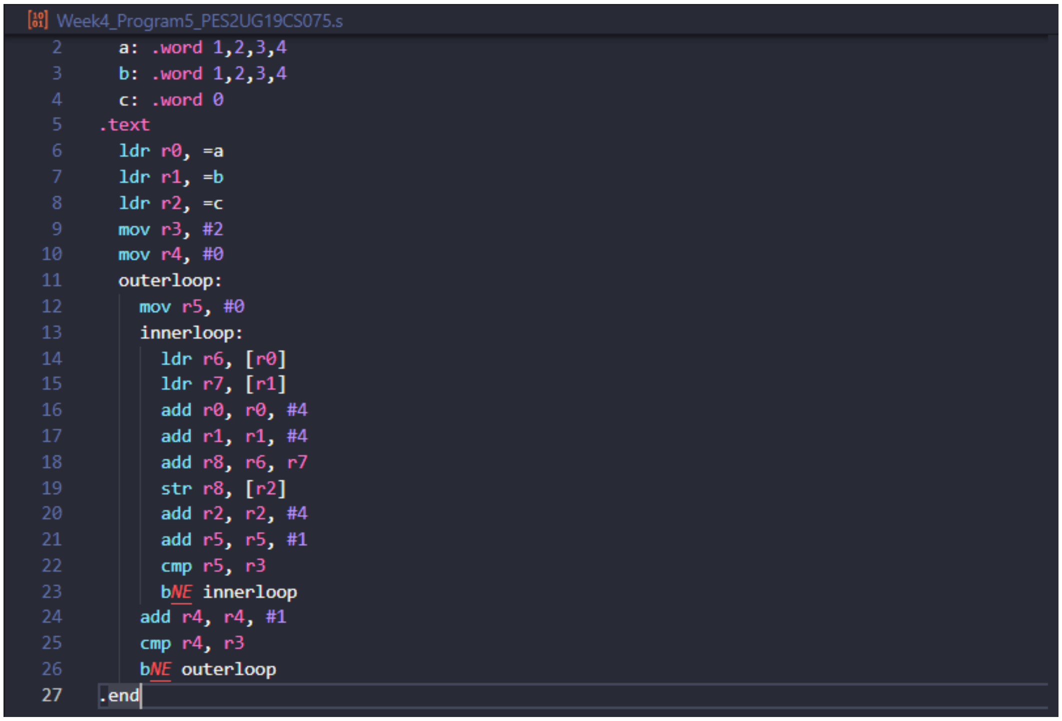
Before execution	a: .word 1,2,3,4	b: .word 0
After Execution	00000001	00000001
	00000002	00000002
	00000003	00000003
	00000004	00000004

Week#4

Program Number: 5

Write an ALP to implement C[i][j]=a[i][j]+b[i][j]

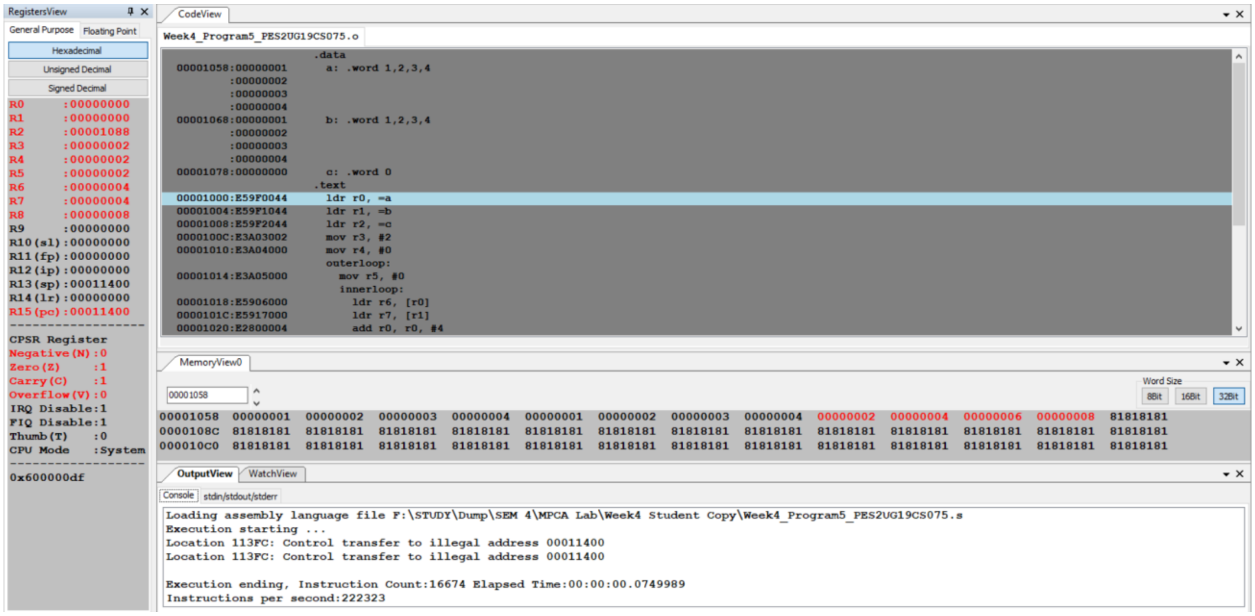
I. ARM Assembly Code (1).



Week4_Program5_PES2UG19CS075.s

```
  a: .word 1,2,3,4
  b: .word 1,2,3,4
  c: .word 0
.text
  ldr r0, =a
  ldr r1, =b
  ldr r2, =c
  mov r3, #2
  mov r4, #0
  outerloop:
    mov r5, #0
    innerloop:
      ldr r6, [r0]
      ldr r7, [r1]
      add r0, r0, #4
      add r1, r1, #4
      add r8, r6, r7
      str r8, [r2]
      add r2, r2, #4
      add r5, r5, #1
      cmp r5, r3
      bNE innerloop
    add r4, r4, #1
    cmp r4, r3
    bNE outerloop
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0        :00000000
R1        :00000000
R2        :00001088
R3        :00000002
R4        :00000002
R5        :00000002
R6        :00000004
R7        :00000004
R8        :00000008
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00011400
R14(lr)   :00000000
R15(pc)   :00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

Week4_Program5_PES2UG19CS075.o

```
                              .data
  00001058:00000001        a: .word 1,2,3,4
          :00000002
          :00000003
          :00000004
  00001068:00000001        b: .word 1,2,3,4
          :00000002
          :00000003
          :00000004
  00001078:00000000        c: .word 0
                              .text
  00001000:E59F0044        ldr r0, =a
  00001004:E59F1044        ldr r1, =b
  00001008:E59F2044        ldr r2, =c
  0000100C:E3A03002        mov r3, #2
  00001010:E3A04000        mov r4, #0
                           outerloop:
  00001014:E3A05000          mov r5, #0
                             innerloop:
  00001018:E5906000            ldr r6, [r0]
  0000101C:E5917000            ldr r7, [r1]
  00001020:E2800004            add r0, r0, #4
```

00001058

```
00001058  00000001  00000002  00000003  00000004  00000001  00000002  00000003  00000004  00000002  00000004  00000006  00000008  81818181
0000108C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010C0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program5_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16674 Elapsed Time:00:00:00.0749989
Instructions per second:222323
```

III. Output Table for the program(1)

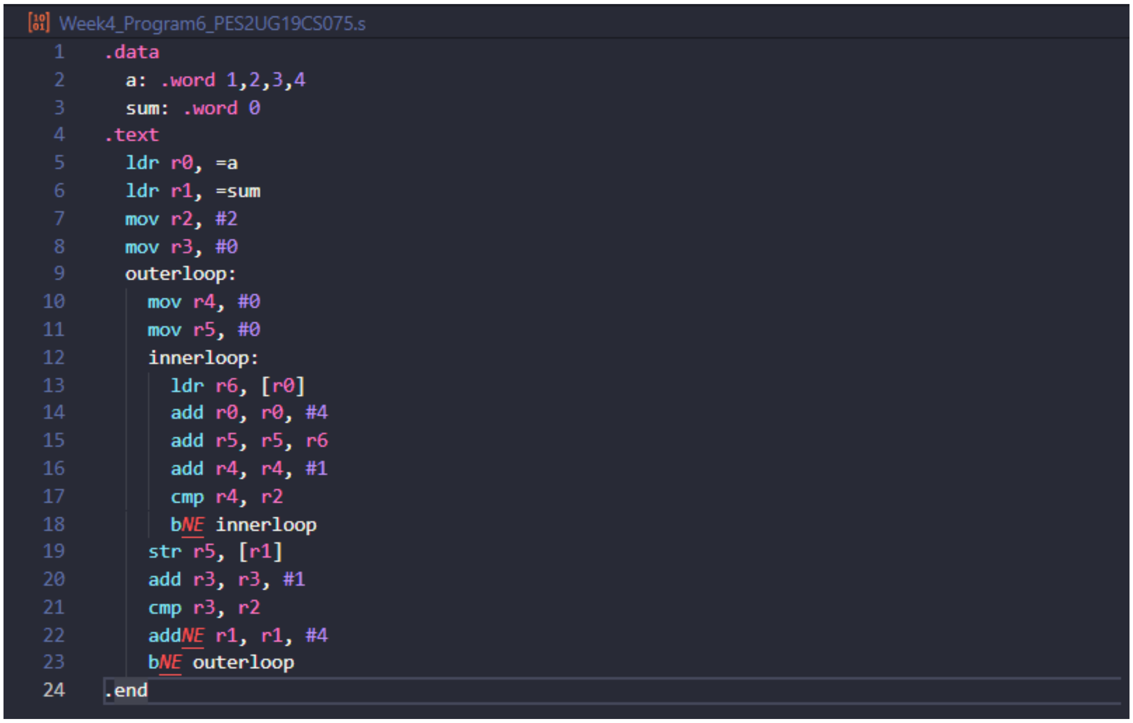
Before execution	a:.word 1,2,3,4	b:.word 1,2,3,4	c:.word 0
After Execution	00000001	00000001	00000002
	00000002	00000002	00000004
	00000003	00000003	00000006
	00000004	00000004	00000008

Week#4

Program Number: 6

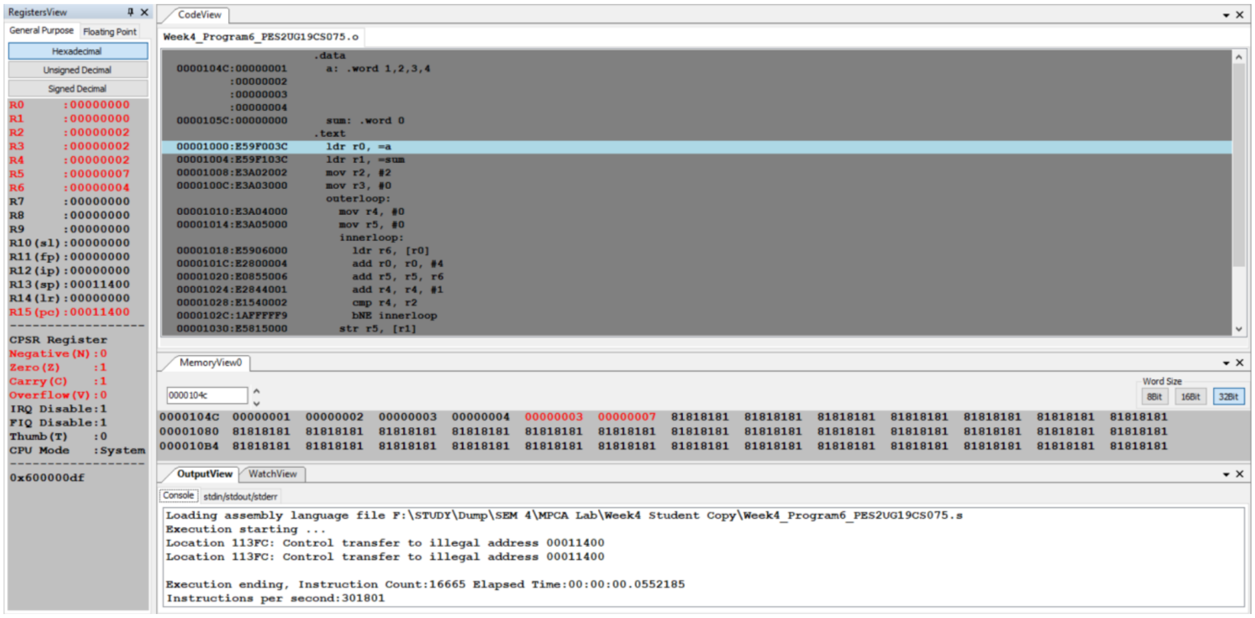
Write an ALP to implement Sum[i] +=a[i][j]

I. ARM Assembly Code (1).



```
.data
  a: .word 1,2,3,4
  sum: .word 0
.text
  ldr r0, =a
  ldr r1, =sum
  mov r2, #2
  mov r3, #0
  outerloop:
    mov r4, #0
    mov r5, #0
    innerloop:
      ldr r6, [r0]
      add r0, r0, #4
      add r5, r5, r6
      add r4, r4, #1
      cmp r4, r2
      bNE innerloop
    str r5, [r1]
    add r3, r3, #1
    cmp r3, r2
    addNE r1, r1, #4
    bNE outerloop
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0        :00000000
R1        :00000000
R2        :00000002
R3        :00000002
R4        :00000002
R5        :00000007
R6        :00000004
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
Week4_Program6_PES2UG19CS075.o

                          .data
 0000104C:00000001          a: .word 1,2,3,4
         :00000002
         :00000003
         :00000004
 0000105C:00000000          sum: .word 0
                          .text
 00001000:E59F003C          ldr r0, =a
 00001004:E59F103C          ldr r1, =sum
 00001008:E3A02002          mov r2, #2
 0000100C:E3A03000          mov r3, #0
                            outerloop:
 00001010:E3A04000            mov r4, #0
 00001014:E3A05000            mov r5, #0
                              innerloop:
 00001018:E5906000              ldr r6, [r0]
 0000101C:E2800004              add r0, r0, #4
 00001020:E0855006              add r5, r5, r6
 00001024:E2844001              add r4, r4, #1
 00001028:E1540002              cmp r4, r2
 0000102C:1AFFFFF9              bNE innerloop
 00001030:E5815000            str r5, [r1]
```

```
0000104c
0000104C  00000001  00000002  00000003  00000004  00000003  00000007  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001080  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Week4 Student Copy\Week4_Program6_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16665 Elapsed Time:00:00:00.0552185
Instructions per second:301801
```

III. Output Table for the program(1)

Before execution	a:.word 1,2,3,4		
After Execution	Addition result	Sum[0]=3	Sum[1]=7

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: *Atul Anurag*

Name: Atul Anurag

SRN: PES2UG19CS075

Section: B

Date: 19-02-2021