ARM7TDMI PROGRAMMING

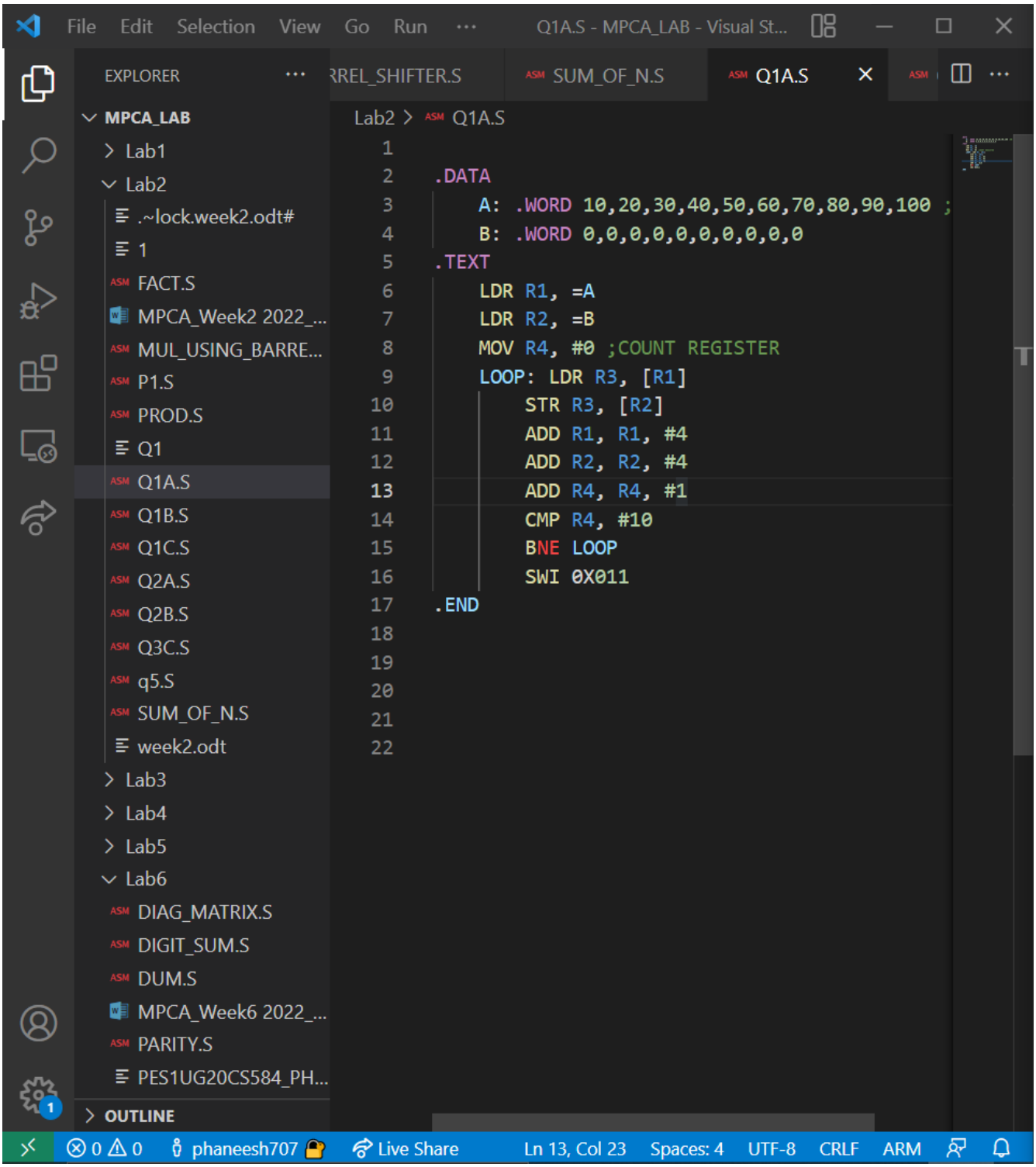
WEEK-2

PHANEESH
PES1UG20CS584
J SECTION

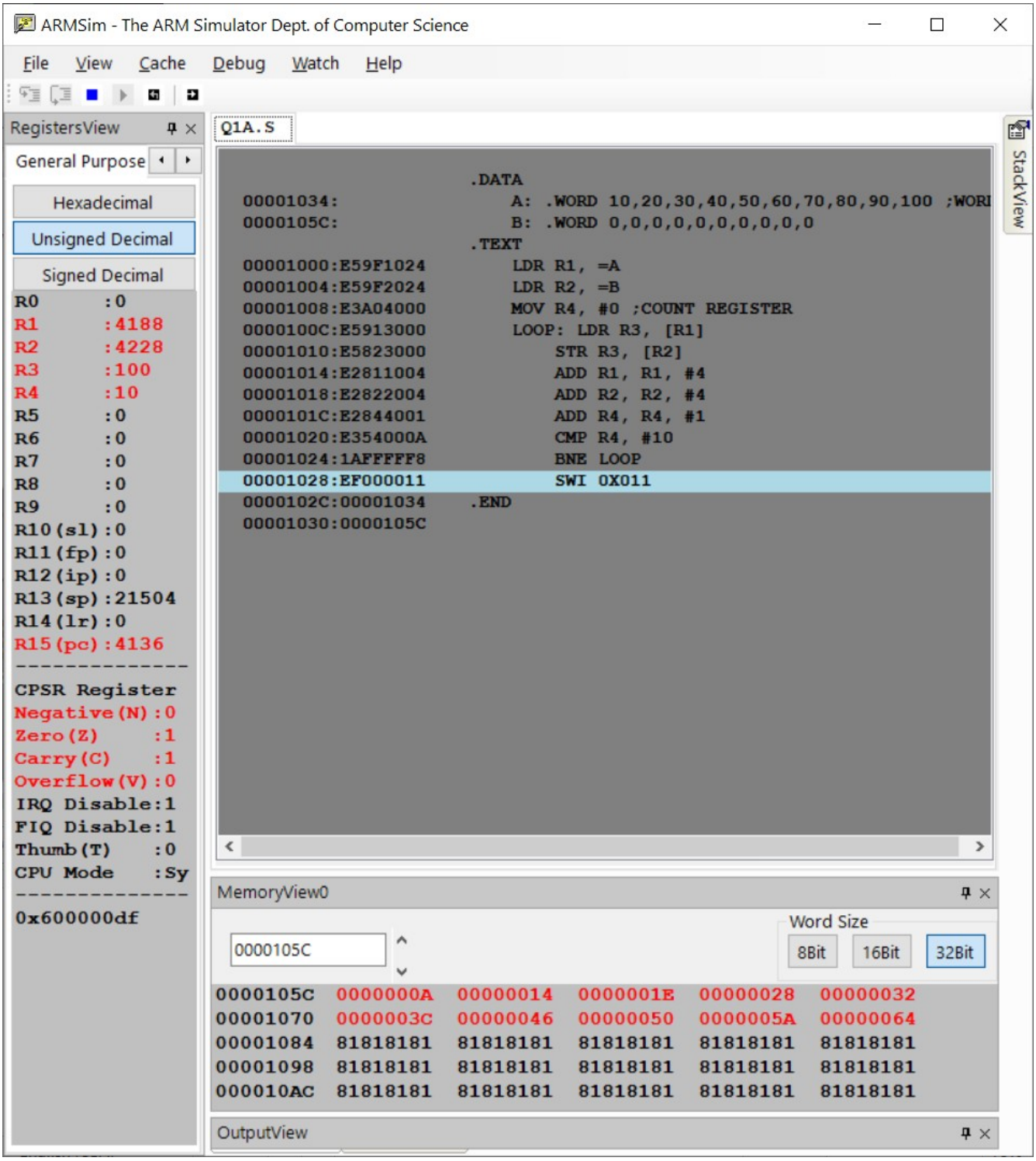
1. Write a program in ARM7TDMI-ISA to copy a block of N data items from Location A to Location B.

a) Use Full word (.word directive)

CODE:

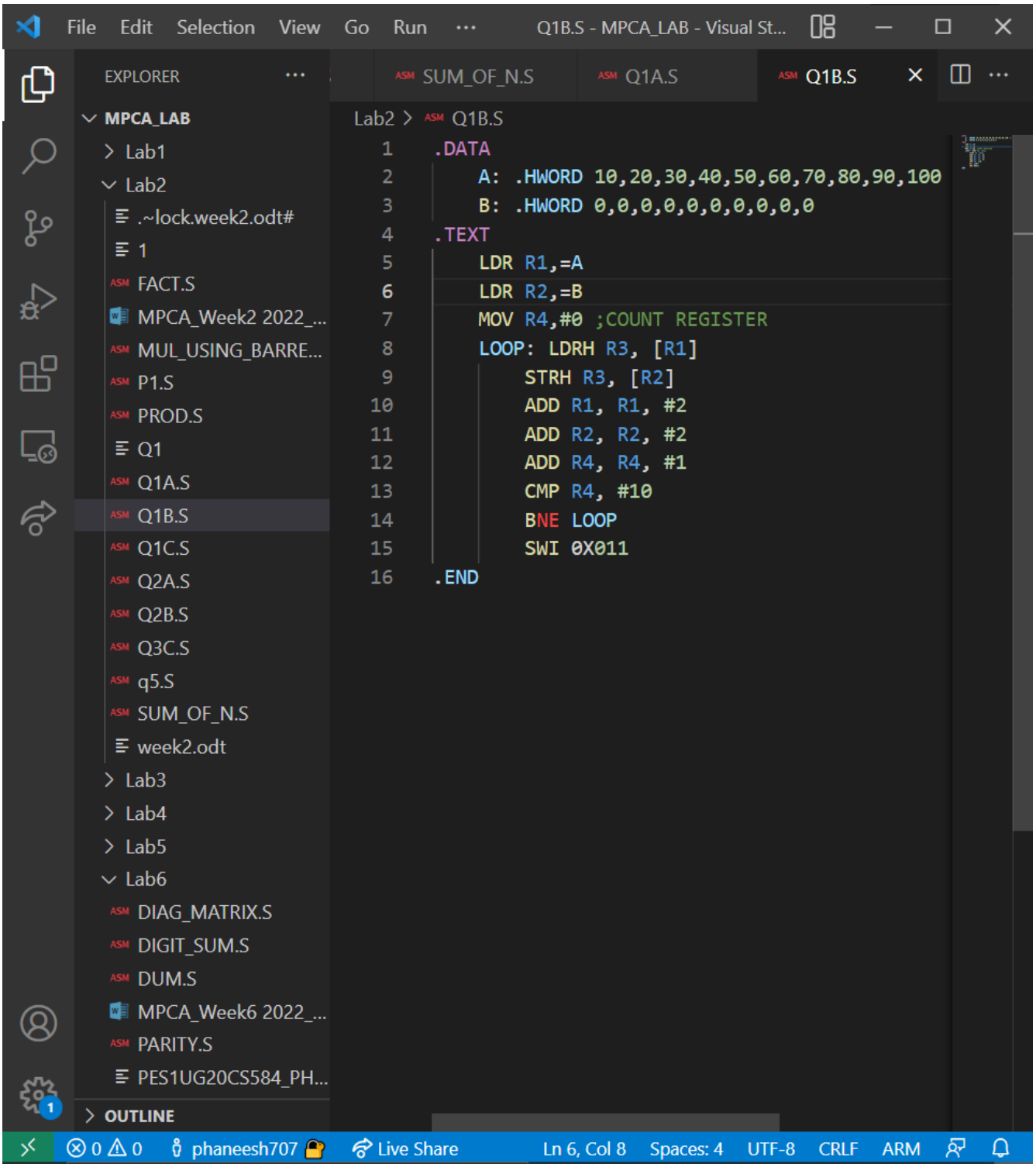


OUTPUT:

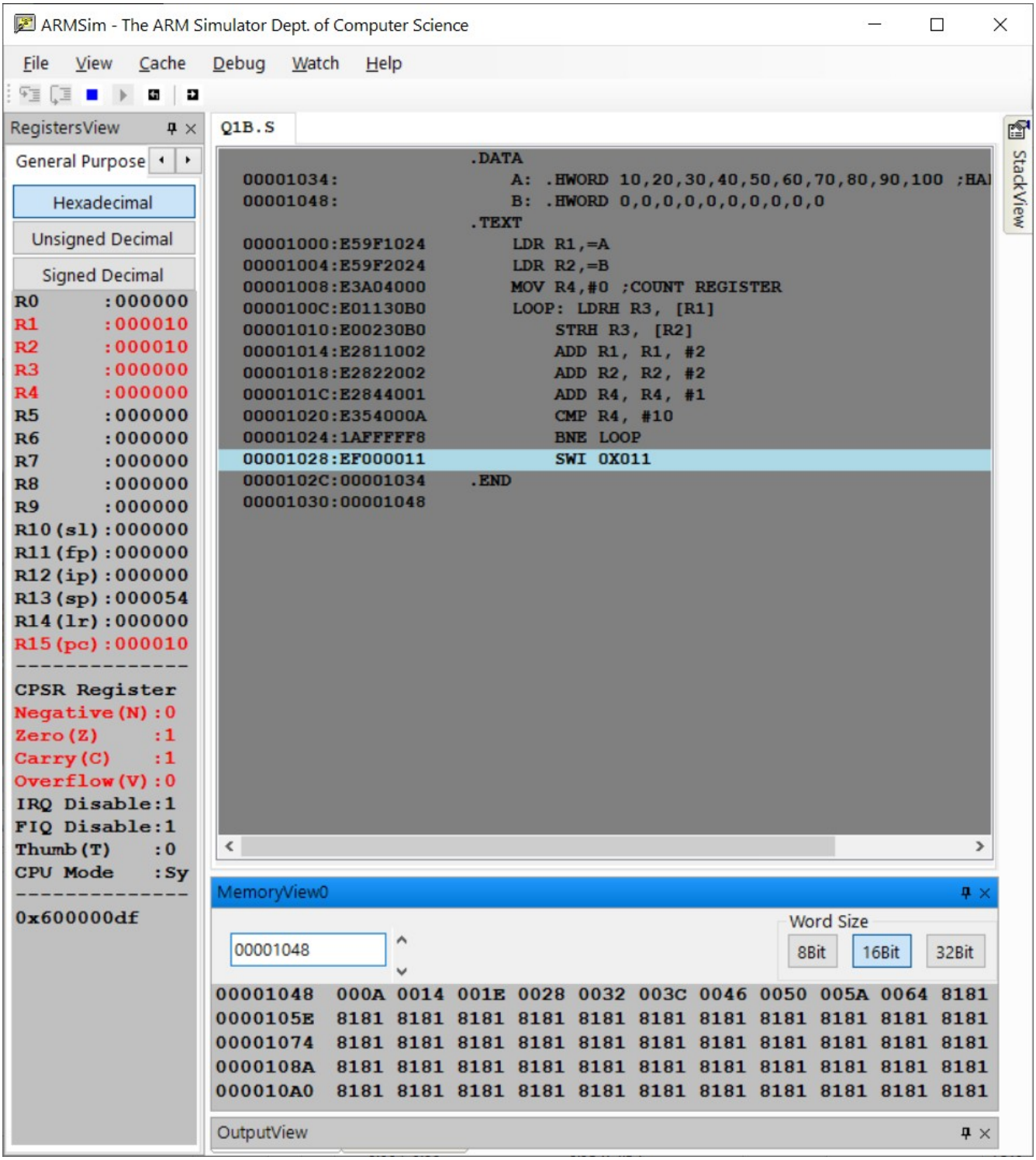

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0
R1 :4188
R2 :4228
R3 :100
R4 :10
R5 :0
R6 :0
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4136
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x600000df
Q1A.S
.DATA
00001034: A: .WORD 10,20,30,40,50,60,70,80,90,100 ;WORI
0000105C: B: .WORD 0,0,0,0,0,0,0,0,0,0
.TEXT
00001000:E59F1024 LDR R1, =A
00001004:E59F2024 LDR R2, =B
00001008:E3A04000 MOV R4, #0 ;COUNT REGISTER
0000100C:E5913000 LOOP: LDR R3, [R1]
00001010:E5823000 STR R3, [R2]
00001014:E2811004 ADD R1, R1, #4
00001018:E2822004 ADD R2, R2, #4
0000101C:E2844001 ADD R4, R4, #1
00001020:E354000A CMP R4, #10
00001024:1AFFFFF8 BNE LOOP
00001028:EF000011 SWI 0X011
0000102C:00001034 .END
00001030:0000105C
StackView
MemoryView0
0000105C
Word Size
8Bit
16Bit
32Bit
0000105C 0000000A 00000014 0000001E 00000028 00000032
00001070 0000003C 00000046 00000050 0000005A 00000064
00001084 81818181 81818181 81818181 81818181 81818181
00001098 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181
OutputView

b) Use Half word(.Hword directive)

CODE:

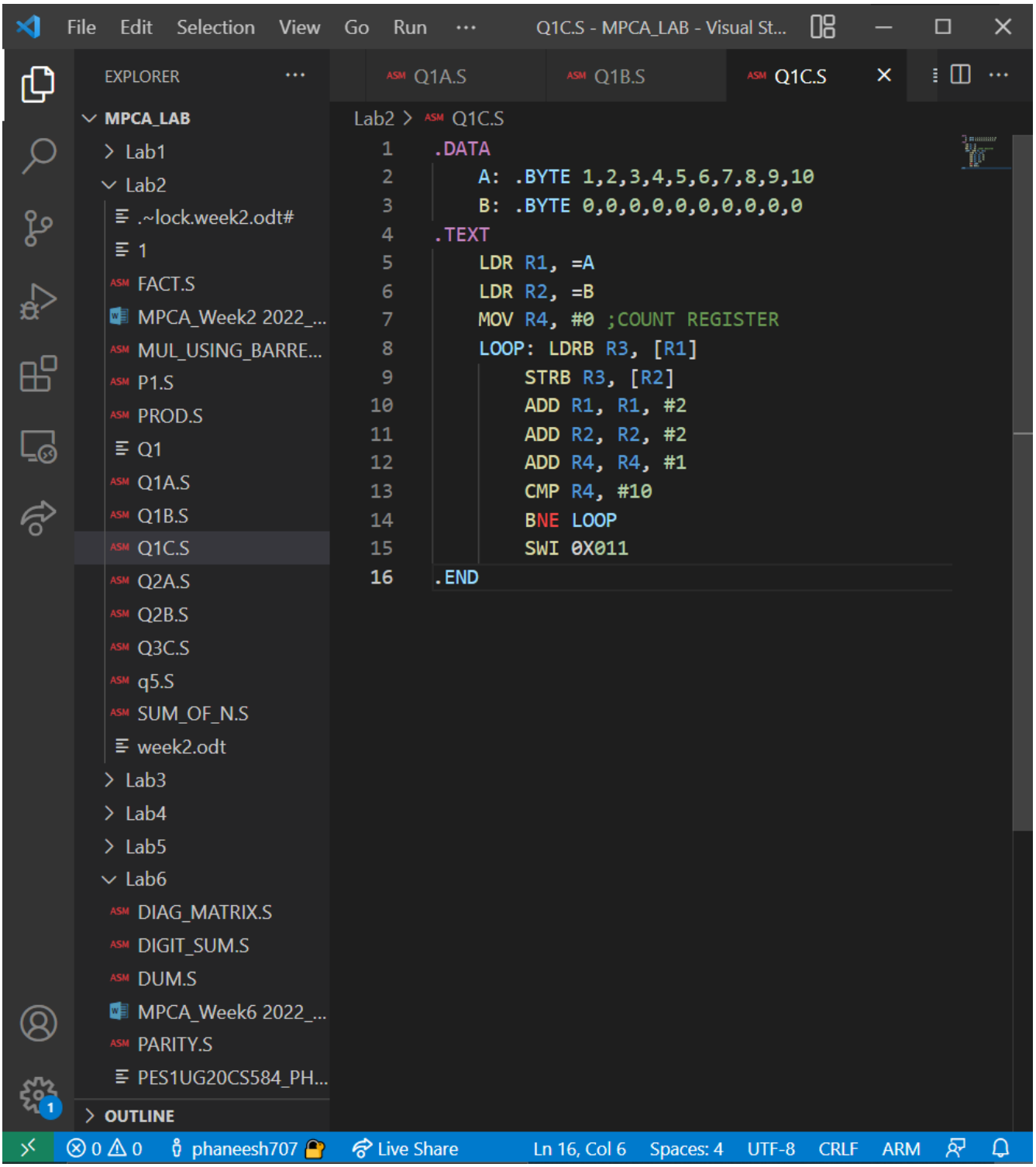


OUTPUT:

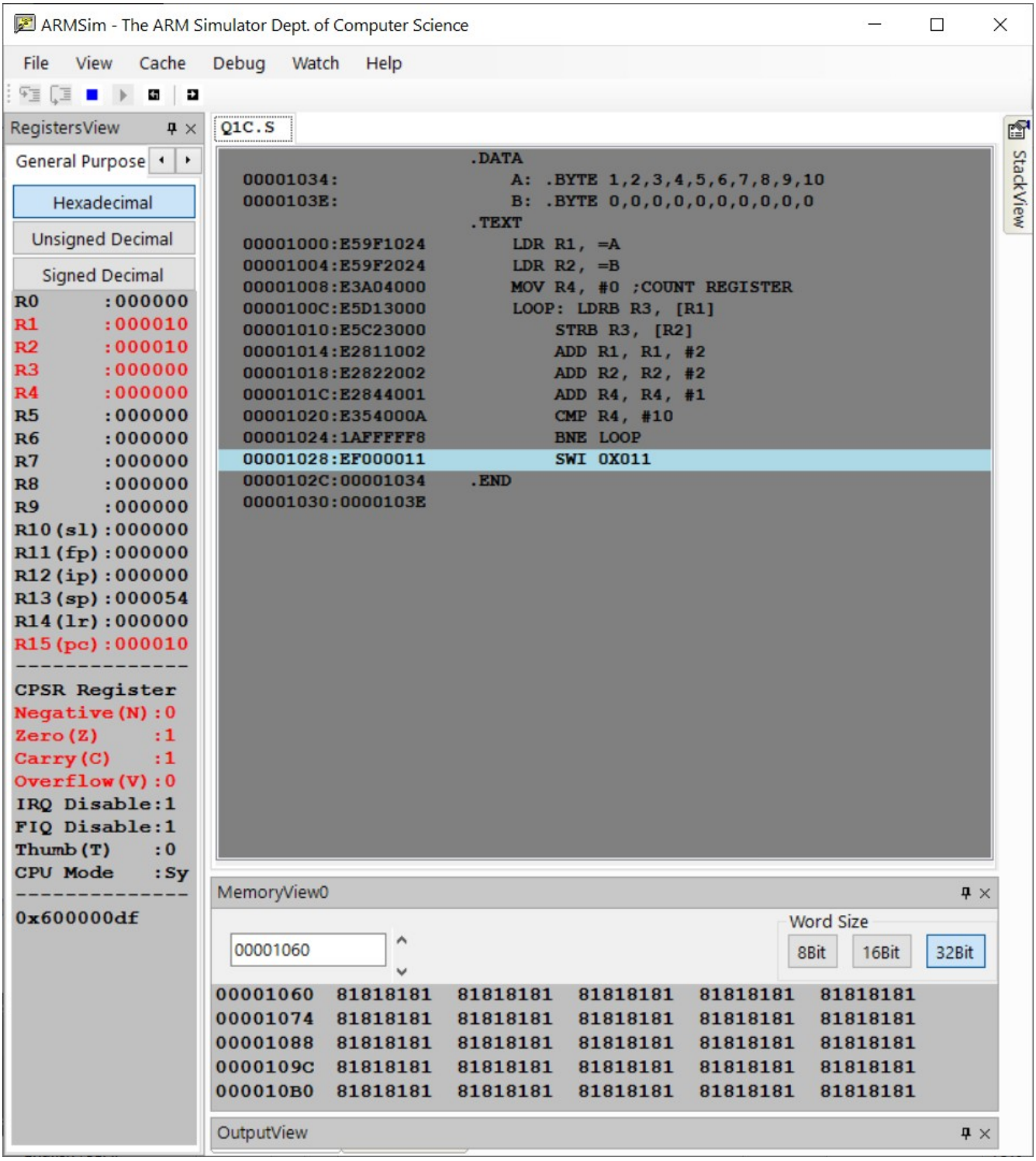

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000000
R1 :000010
R2 :000010
R3 :000000
R4 :000000
R5 :000000
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10(sl):000000
R11(fp):000000
R12(ip):000000
R13(sp):000054
R14(lr):000000
R15(pc):000010
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 :Sy
0x600000df
Q1B.S
.DATA
00001034: A: .HWORD 10,20,30,40,50,60,70,80,90,100 ;HA
00001048: B: .HWORD 0,0,0,0,0,0,0,0,0,0
.TEXT
00001000:E59F1024 LDR R1,=A
00001004:E59F2024 LDR R2,=B
00001008:E3A04000 MOV R4,#0 ;COUNT REGISTER
0000100C:E01130B0 LOOP: LDRH R3, [R1]
00001010:E00230B0 STRH R3, [R2]
00001014:E2811002 ADD R1, R1, #2
00001018:E2822002 ADD R2, R2, #2
0000101C:E2844001 ADD R4, R4, #1
00001020:E354000A CMP R4, #10
00001024:1AFFFFF8 BNE LOOP
00001028:EF000011 SWI 0X011
0000102C:00001034 .END
00001030:00001048
StackView
MemoryView0
00001048
Word Size
8Bit
16Bit
32Bit
00001048 000A 0014 001E 0028 0032 003C 0046 0050 005A 0064 8181
0000105E 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001074 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000108A 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010A0 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
OutputView

c) Use Byte wise (.Byte directive)

CODE:



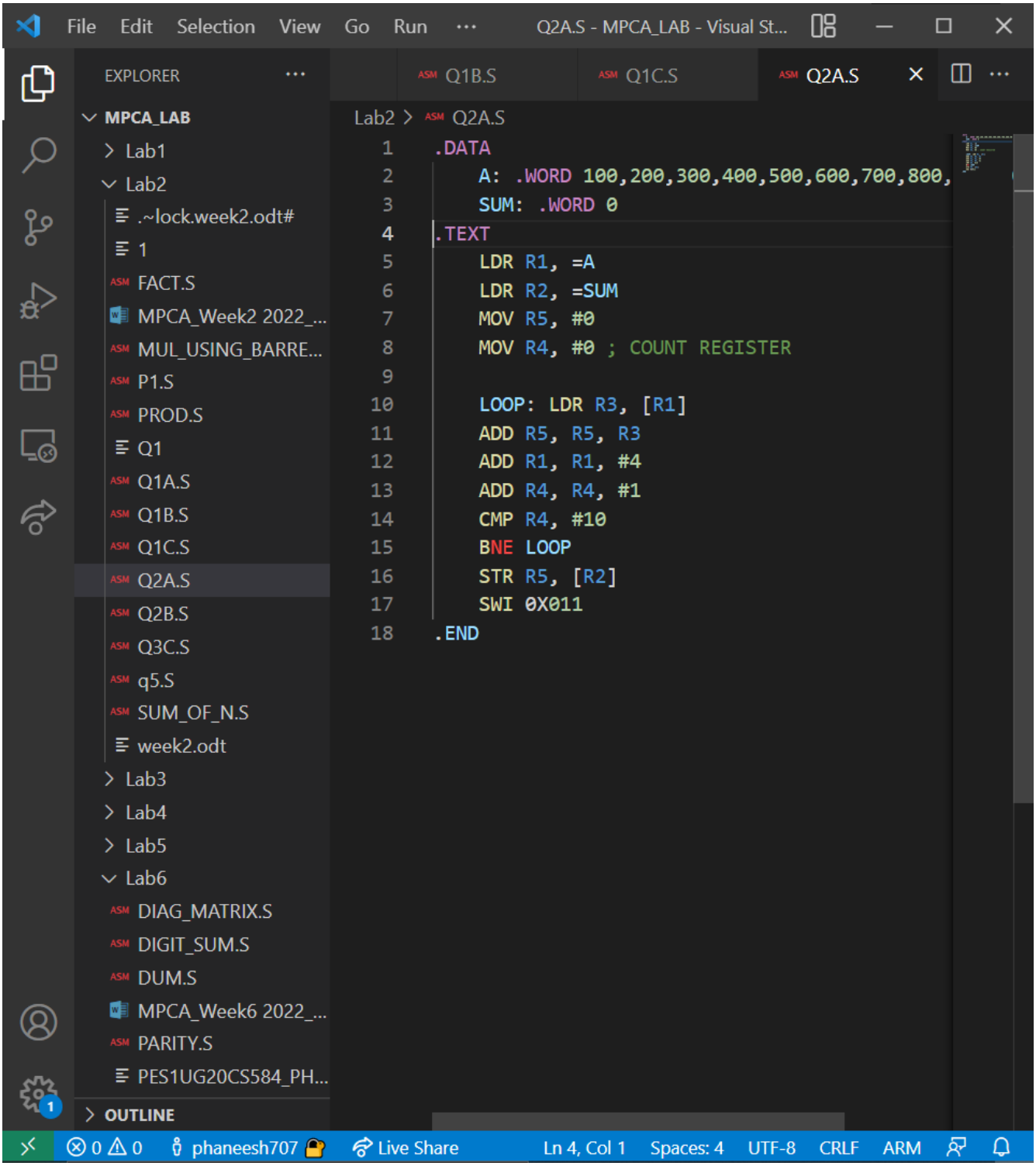
OUTPUT:


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000000
R1 :000010
R2 :000010
R3 :000000
R4 :000000
R5 :000000
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10(sl):000000
R11(fp):000000
R12(ip):000000
R13(sp):000054
R14(lr):000000
R15(pc):000010
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 :Sy
0x600000df
Q1C.S
.DATA
00001034: A: .BYTE 1,2,3,4,5,6,7,8,9,10
0000103E: B: .BYTE 0,0,0,0,0,0,0,0,0,0
.TEXT
00001000:E59F1024 LDR R1, =A
00001004:E59F2024 LDR R2, =B
00001008:E3A04000 MOV R4, #0 ;COUNT REGISTER
0000100C:E5D13000 LOOP: LDRB R3, [R1]
00001010:E5C23000 STRB R3, [R2]
00001014:E2811002 ADD R1, R1, #2
00001018:E2822002 ADD R2, R2, #2
0000101C:E2844001 ADD R4, R4, #1
00001020:E354000A CMP R4, #10
00001024:1AFFFFF8 BNE LOOP
00001028:EF000011 SWI 0X011
0000102C:00001034 .END
00001030:0000103E
StackView
MemoryView0
00001060
Word Size
8Bit 16Bit 32Bit
00001060 81818181 81818181 81818181 81818181 81818181
00001074 81818181 81818181 81818181 81818181 81818181
00001088 81818181 81818181 81818181 81818181 81818181
0000109C 81818181 81818181 81818181 81818181 81818181
000010B0 81818181 81818181 81818181 81818181 81818181
OutputView

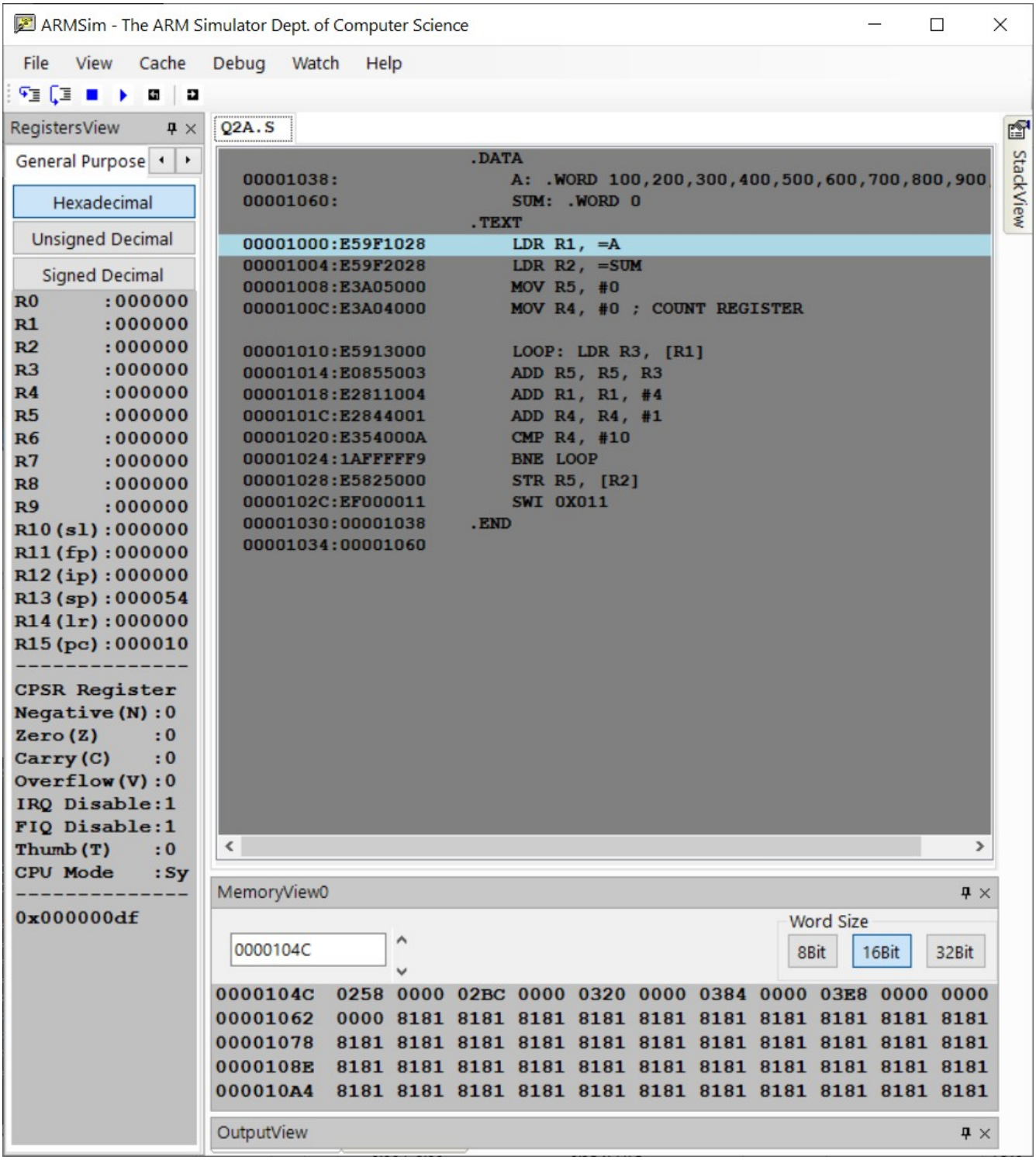
2. Write a program in ARM7TDMI-ISA to find the sum of N data items in the memory. Store the result in the memory location.

a) Use Full word (.word directive)

CODE:

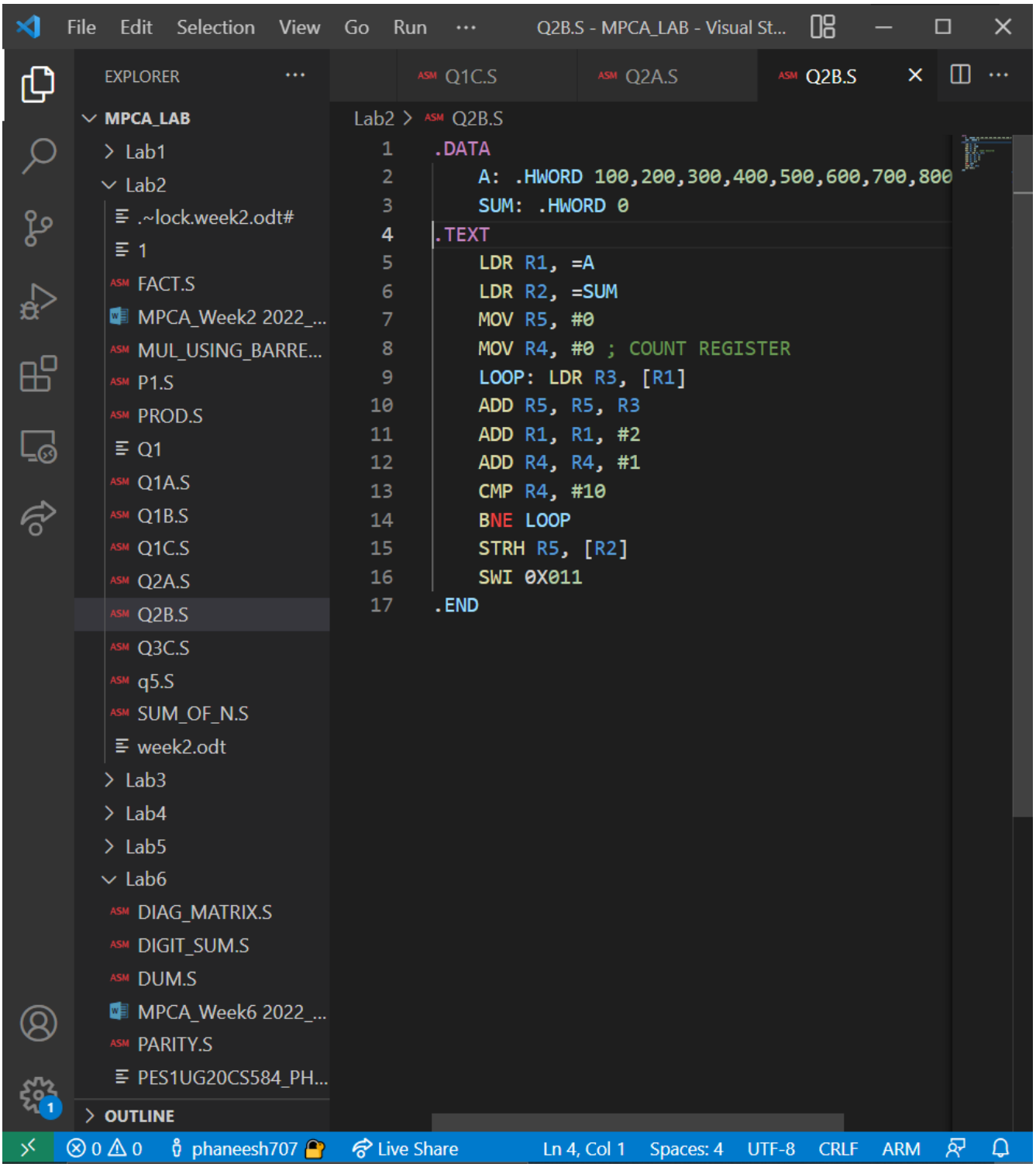


OUTPUT:

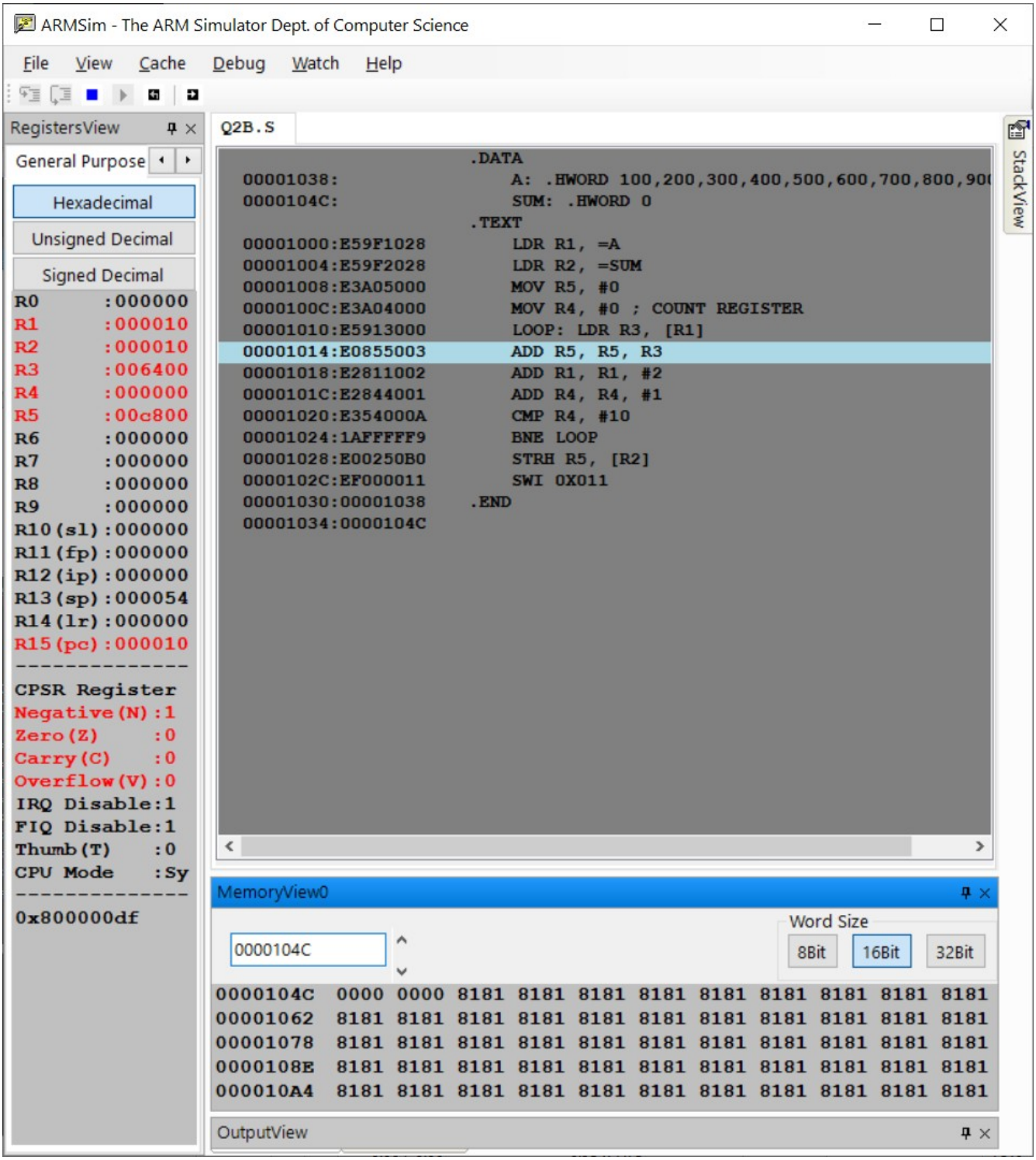

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000000
R1 :000000
R2 :000000
R3 :000000
R4 :000000
R5 :000000
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10(sl):000000
R11(fp):000000
R12(ip):000000
R13(sp):000054
R14(lr):000000
R15(pc):000010
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x000000df
Q2A.S
.DATA
00001038: A: .WORD 100,200,300,400,500,600,700,800,900
00001060: SUM: .WORD 0
.TEXT
00001000:E59F1028 LDR R1, =A
00001004:E59F2028 LDR R2, =SUM
00001008:E3A05000 MOV R5, #0
0000100C:E3A04000 MOV R4, #0 ; COUNT REGISTER
00001010:E5913000 LOOP: LDR R3, [R1]
00001014:E0855003 ADD R5, R5, R3
00001018:E2811004 ADD R1, R1, #4
0000101C:E2844001 ADD R4, R4, #1
00001020:E354000A CMP R4, #10
00001024:1AFFFFF9 BNE LOOP
00001028:E5825000 STR R5, [R2]
0000102C:EF000011 SWI 0X011
00001030:00001038 .END
00001034:00001060
StackView
MemoryView0
0000104C
Word Size
8Bit
16Bit
32Bit
0000104C 0258 0000 02BC 0000 0320 0000 0384 0000 03E8 0000 0000
00001062 0000 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001078 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000108E 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010A4 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
OutputView

b) Use Half word(.Hword directive)

CODE:

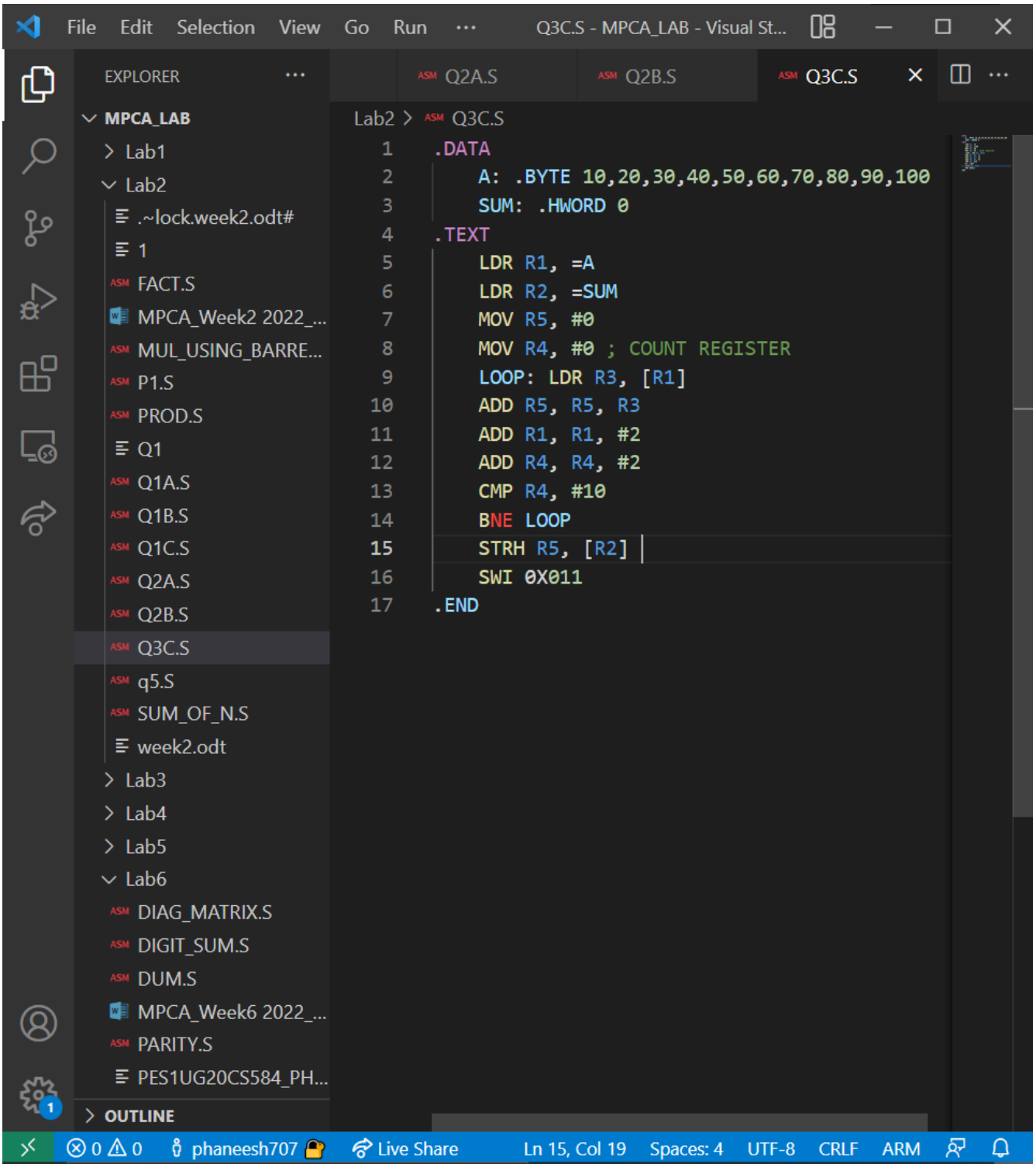


OUTPUT:

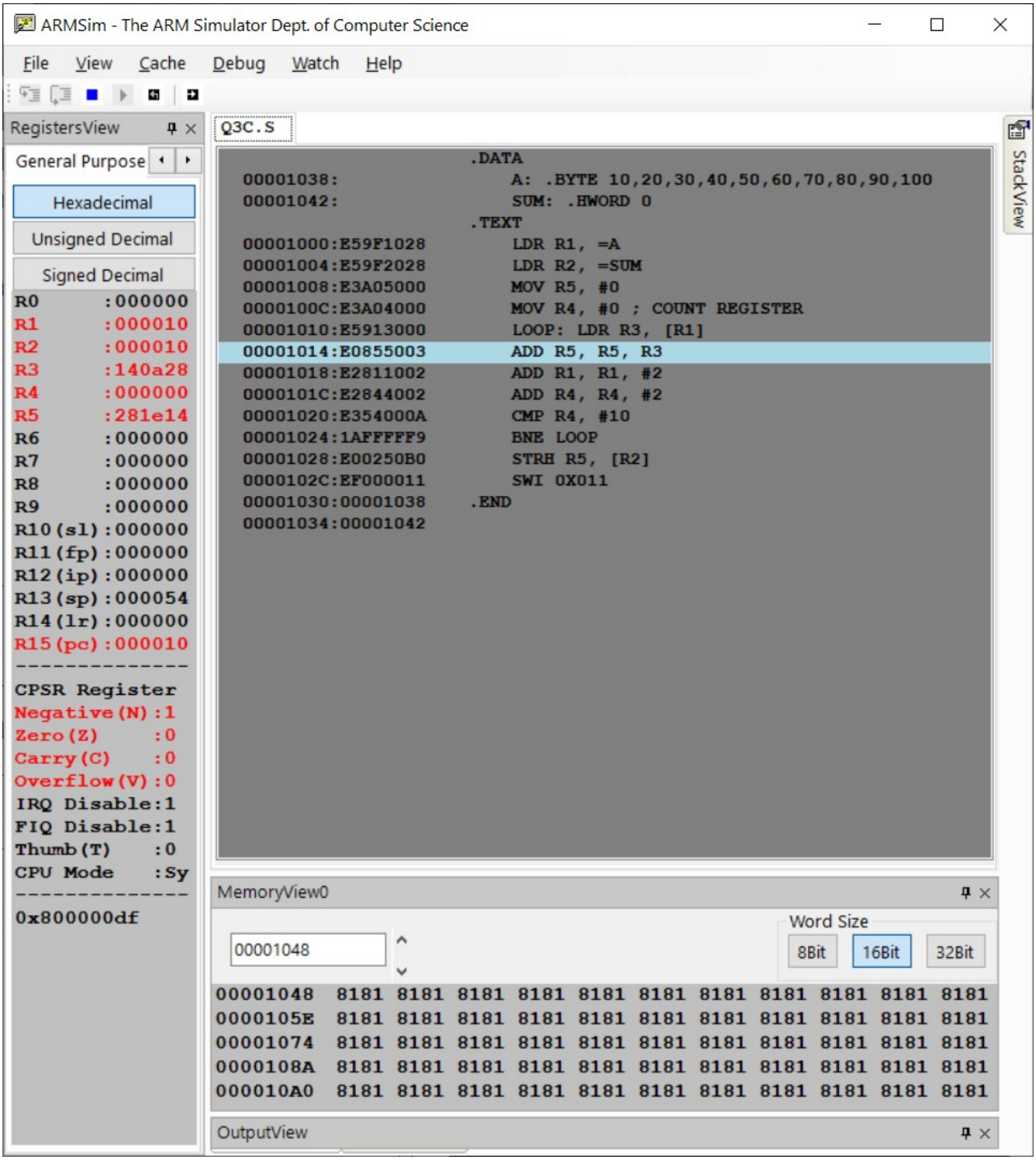

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000000
R1 :000010
R2 :000010
R3 :006400
R4 :000000
R5 :00c800
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10(sl):000000
R11(fp):000000
R12(ip):000000
R13(sp):000054
R14(lr):000000
R15(pc):000010
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x800000df
Q2B.S
.DATA
00001038: A: .HWORD 100,200,300,400,500,600,700,800,90
0000104C: SUM: .HWORD 0
.TEXT
00001000:E59F1028 LDR R1, =A
00001004:E59F2028 LDR R2, =SUM
00001008:E3A05000 MOV R5, #0
0000100C:E3A04000 MOV R4, #0 ; COUNT REGISTER
00001010:E5913000 LOOP: LDR R3, [R1]
00001014:E0855003 ADD R5, R5, R3
00001018:E2811002 ADD R1, R1, #2
0000101C:E2844001 ADD R4, R4, #1
00001020:E354000A CMP R4, #10
00001024:1AFFFFF9 BNE LOOP
00001028:E00250B0 STRH R5, [R2]
0000102C:EF000011 SWI 0X011
00001030:00001038 .END
00001034:0000104C
StackView
MemoryView0
0000104C
Word Size
8Bit
16Bit
32Bit
0000104C 0000 0000 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001062 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001078 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000108E 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010A4 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
OutputView

c) Use Byte wise (.Byte directive)

CODE:

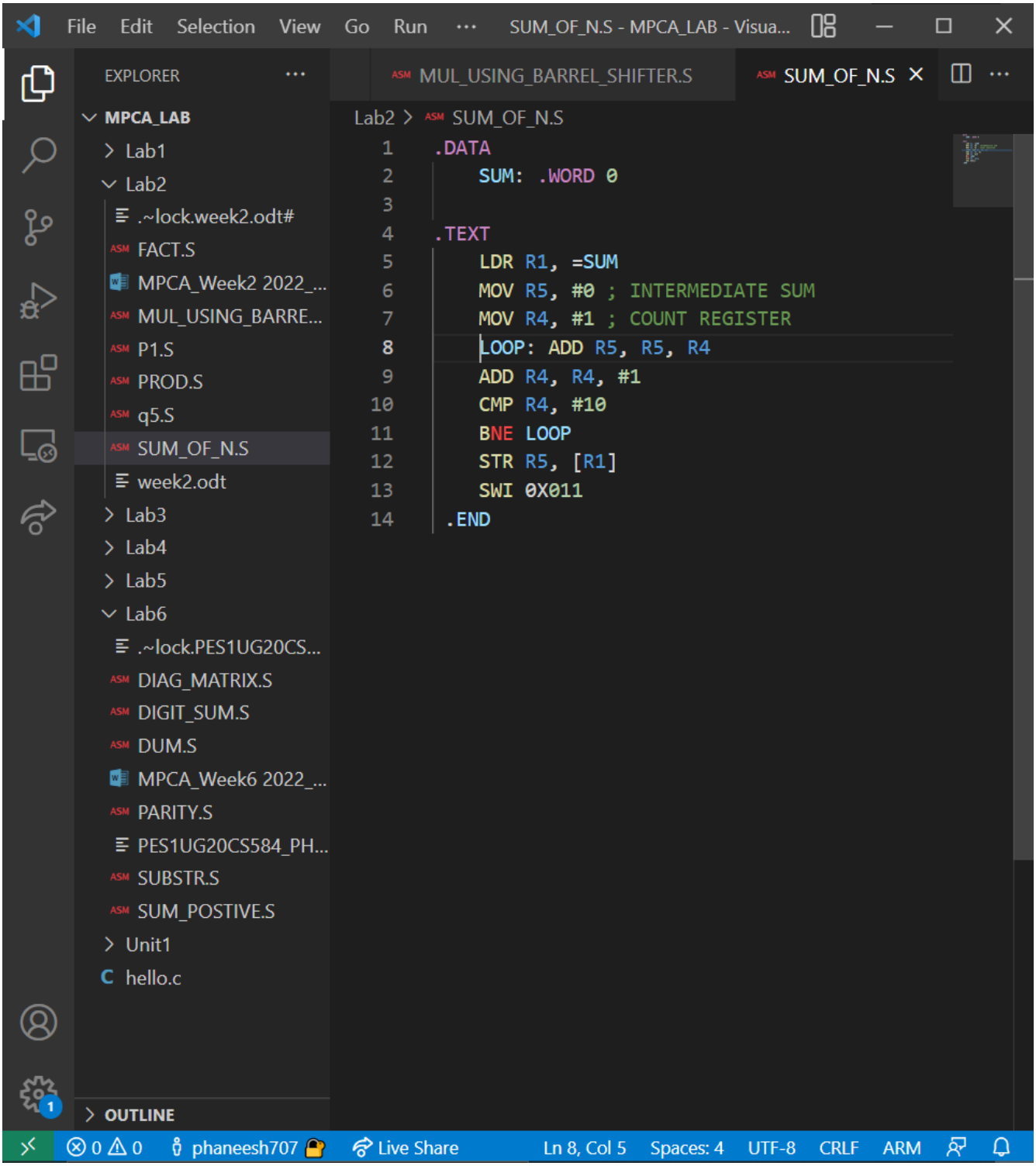


OUTPUT:

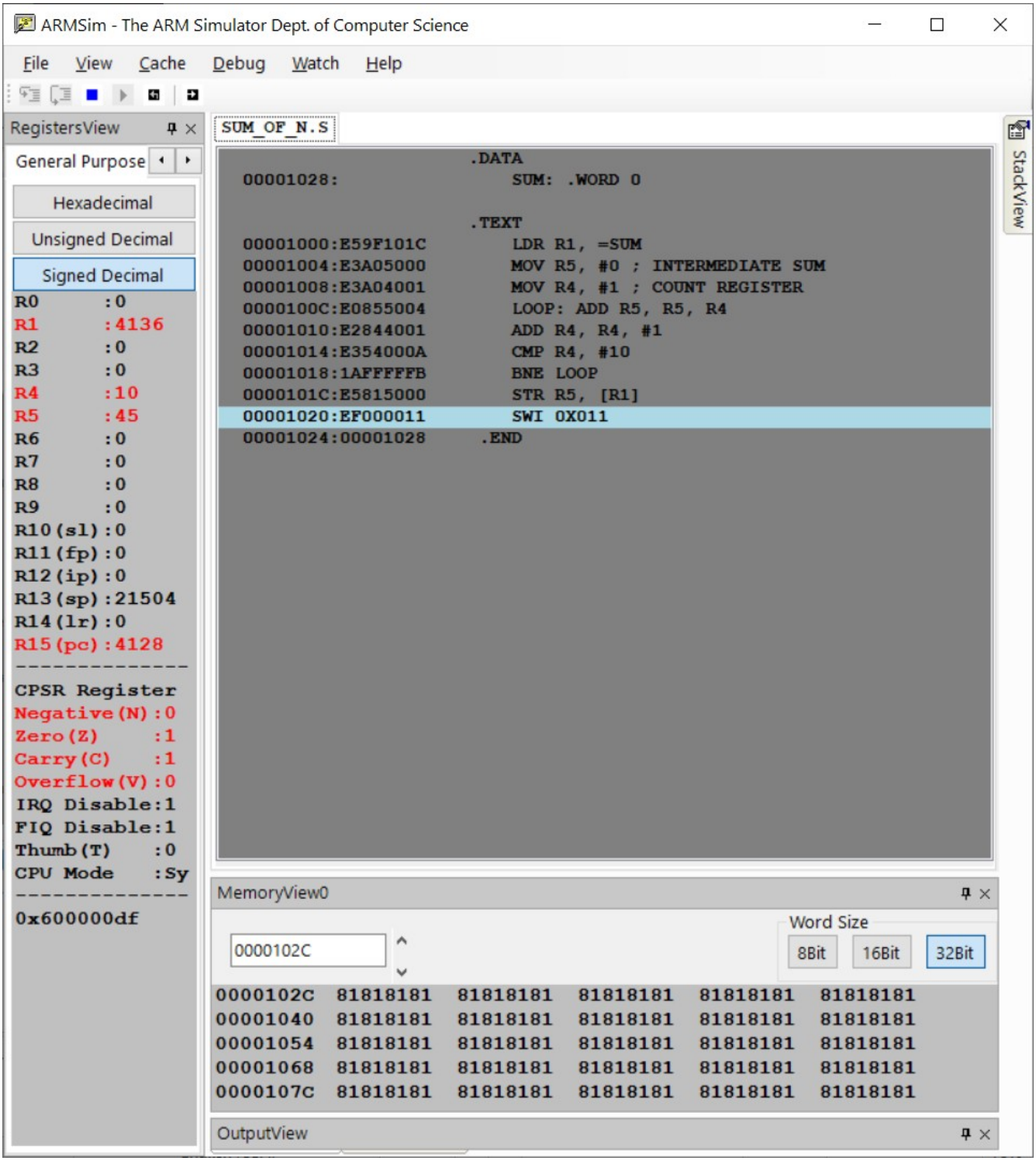

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000000
R1 :000010
R2 :000010
R3 :140a28
R4 :000000
R5 :281e14
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10(sl):000000
R11(fp):000000
R12(ip):000000
R13(sp):000054
R14(lr):000000
R15(pc):000010
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x800000df
Q3C.S
.DATA
00001038: A: .BYTE 10,20,30,40,50,60,70,80,90,100
00001042: SUM: .HWORD 0
.TEXT
00001000:E59F1028 LDR R1, =A
00001004:E59F2028 LDR R2, =SUM
00001008:E3A05000 MOV R5, #0
0000100C:E3A04000 MOV R4, #0 ; COUNT REGISTER
00001010:E5913000 LOOP: LDR R3, [R1]
00001014:E0855003 ADD R5, R5, R3
00001018:E2811002 ADD R1, R1, #2
0000101C:E2844002 ADD R4, R4, #2
00001020:E354000A CMP R4, #10
00001024:1AFFFFF9 BNE LOOP
00001028:E00250B0 STRH R5, [R2]
0000102C:EF000011 SWI 0X011
00001030:00001038 .END
00001034:00001042
StackView
MemoryView0
00001048
Word Size
8Bit
16Bit
32Bit
00001048 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000105E 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001074 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000108A 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010A0 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
OutputView

3. Write a program in ARM7TDMI-ISA to find the sum of N natural numbers. Store the result in the memory location.

CODE:

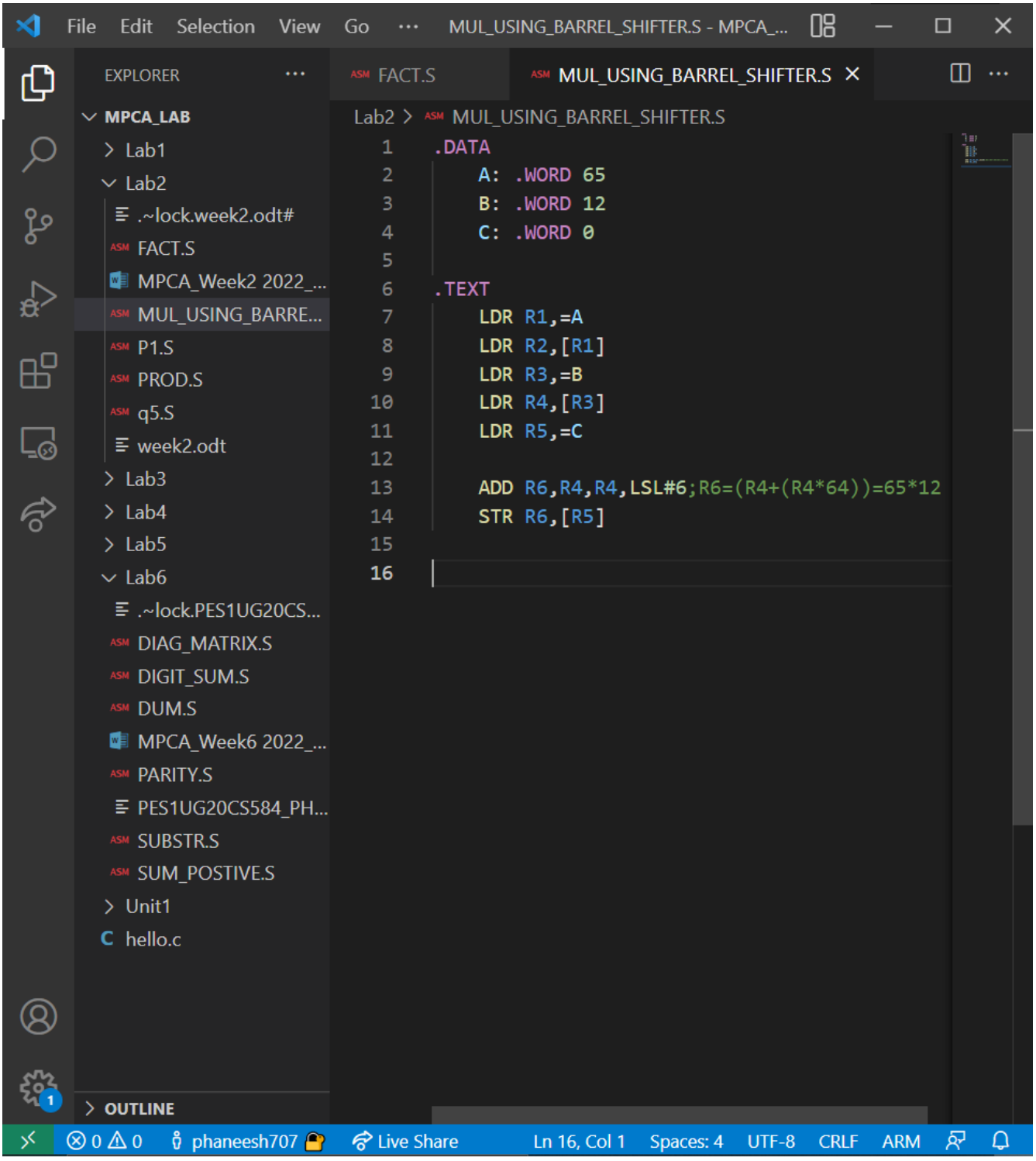


OUTPUT:

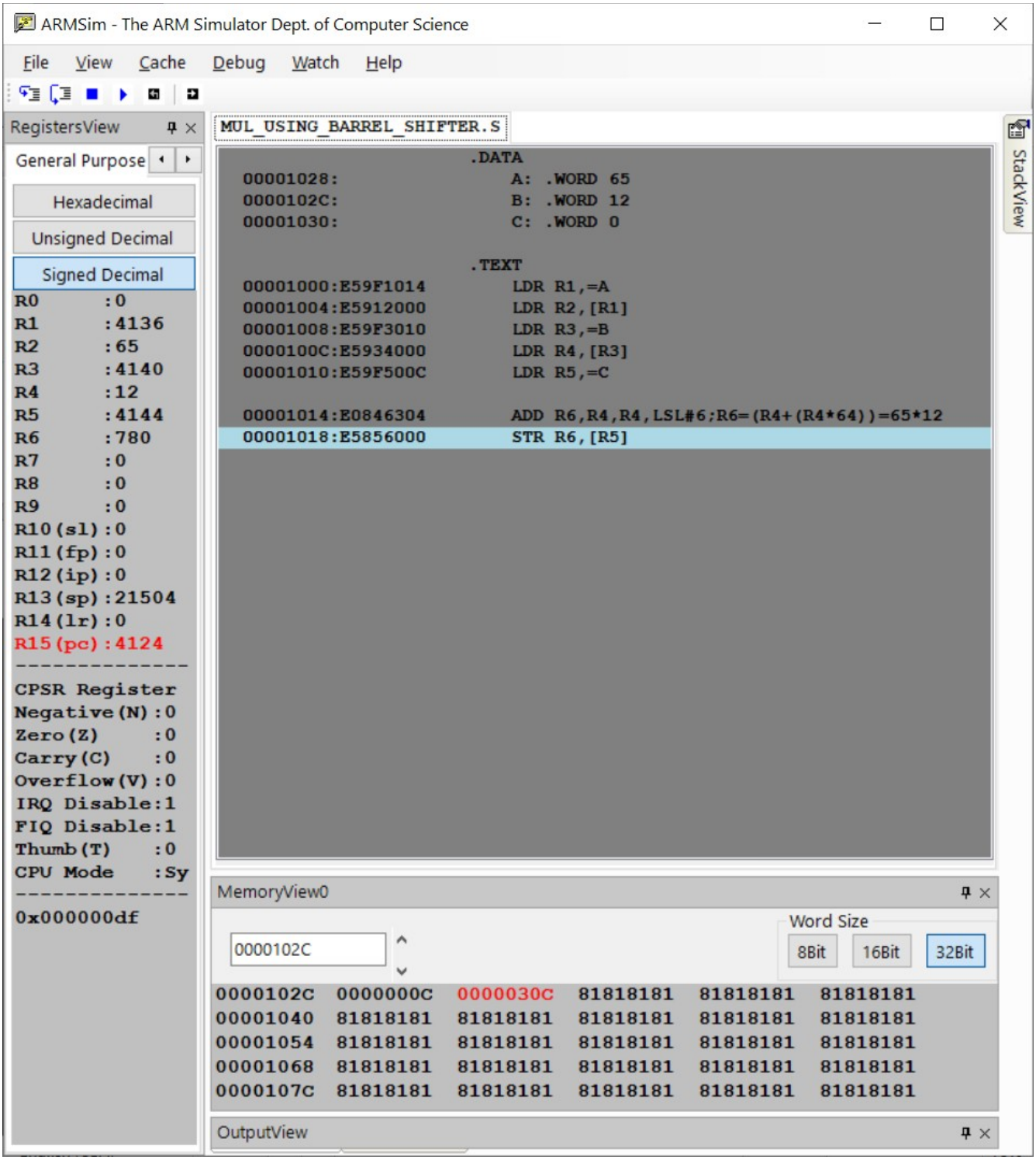

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0
R1 :4136
R2 :0
R3 :0
R4 :10
R5 :45
R6 :0
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4128
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x600000df
SUM_OF_N.S
.DATA
00001028: SUM: .WORD 0
.TEXT
00001000:E59F101C LDR R1, =SUM
00001004:E3A05000 MOV R5, #0 ; INTERMEDIATE SUM
00001008:E3A04001 MOV R4, #1 ; COUNT REGISTER
0000100C:E0855004 LOOP: ADD R5, R5, R4
00001010:E2844001 ADD R4, R4, #1
00001014:E354000A CMP R4, #10
00001018:1AFFFFFB BNE LOOP
0000101C:E5815000 STR R5, [R1]
00001020:EF000011 SWI 0X011
00001024:00001028 .END
StackView
MemoryView0
0000102C
Word Size
8Bit 16Bit 32Bit
0000102C 81818181 81818181 81818181 81818181 81818181
00001040 81818181 81818181 81818181 81818181 81818181
00001054 81818181 81818181 81818181 81818181 81818181
00001068 81818181 81818181 81818181 81818181 81818181
0000107C 81818181 81818181 81818181 81818181 81818181
OutputView

4. Write a program in ARM7TDMI-ISA to find the product of two 32-bit numbers using barrel shifter.

CODE:



OUTPUT:


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0
R1 :4136
R2 :65
R3 :4140
R4 :12
R5 :4144
R6 :780
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4124
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x000000df
MUL_USING_BARREL_SHIFTER.S
.DATA
00001028: A: .WORD 65
0000102C: B: .WORD 12
00001030: C: .WORD 0
.TEXT
00001000:E59F1014 LDR R1,=A
00001004:E5912000 LDR R2,[R1]
00001008:E59F3010 LDR R3,=B
0000100C:E5934000 LDR R4,[R3]
00001010:E59F500C LDR R5,=C
00001014:E0846304 ADD R6,R4,R4,LSL#6;R6=(R4+(R4*64))=65*12
00001018:E5856000 STR R6,[R5]
StackView
MemoryView0
0000102C
Word Size
8Bit 16Bit 32Bit
0000102C 0000000C 0000030C 81818181 81818181 81818181
00001040 81818181 81818181 81818181 81818181 81818181
00001054 81818181 81818181 81818181 81818181 81818181
00001068 81818181 81818181 81818181 81818181 81818181
0000107C 81818181 81818181 81818181 81818181 81818181
OutputView

5. Convert the following statement in C language into an ALP using ARM7TDMI – ISA.

```
IF([A]==[B]) then C=[A]+[B];
     ELSE IF ([B]==[C]) D=[A]-[B];
     ELSE E=[A]*[B]
```

CODE:

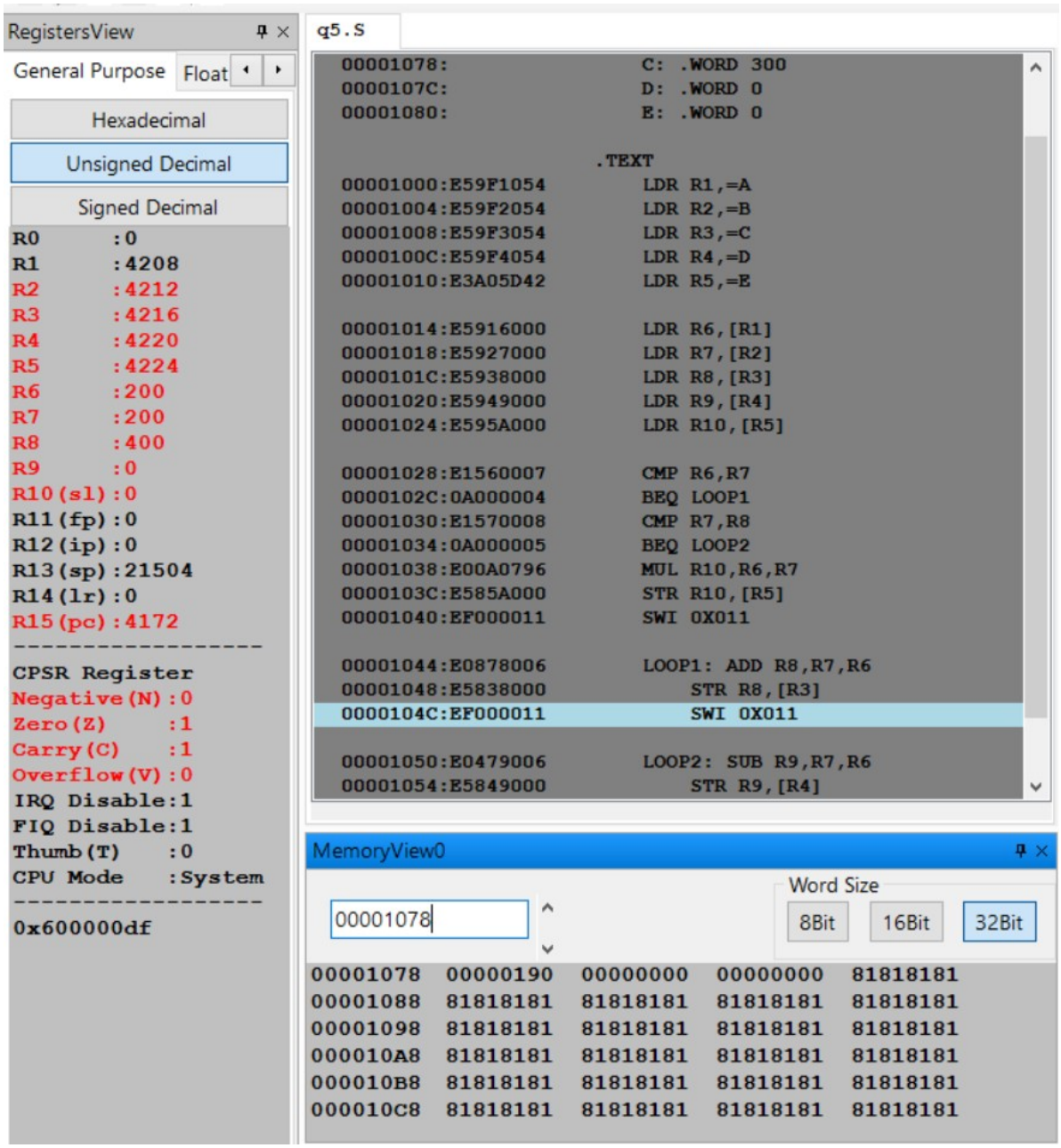
q5.S

Lab2 > q5.S

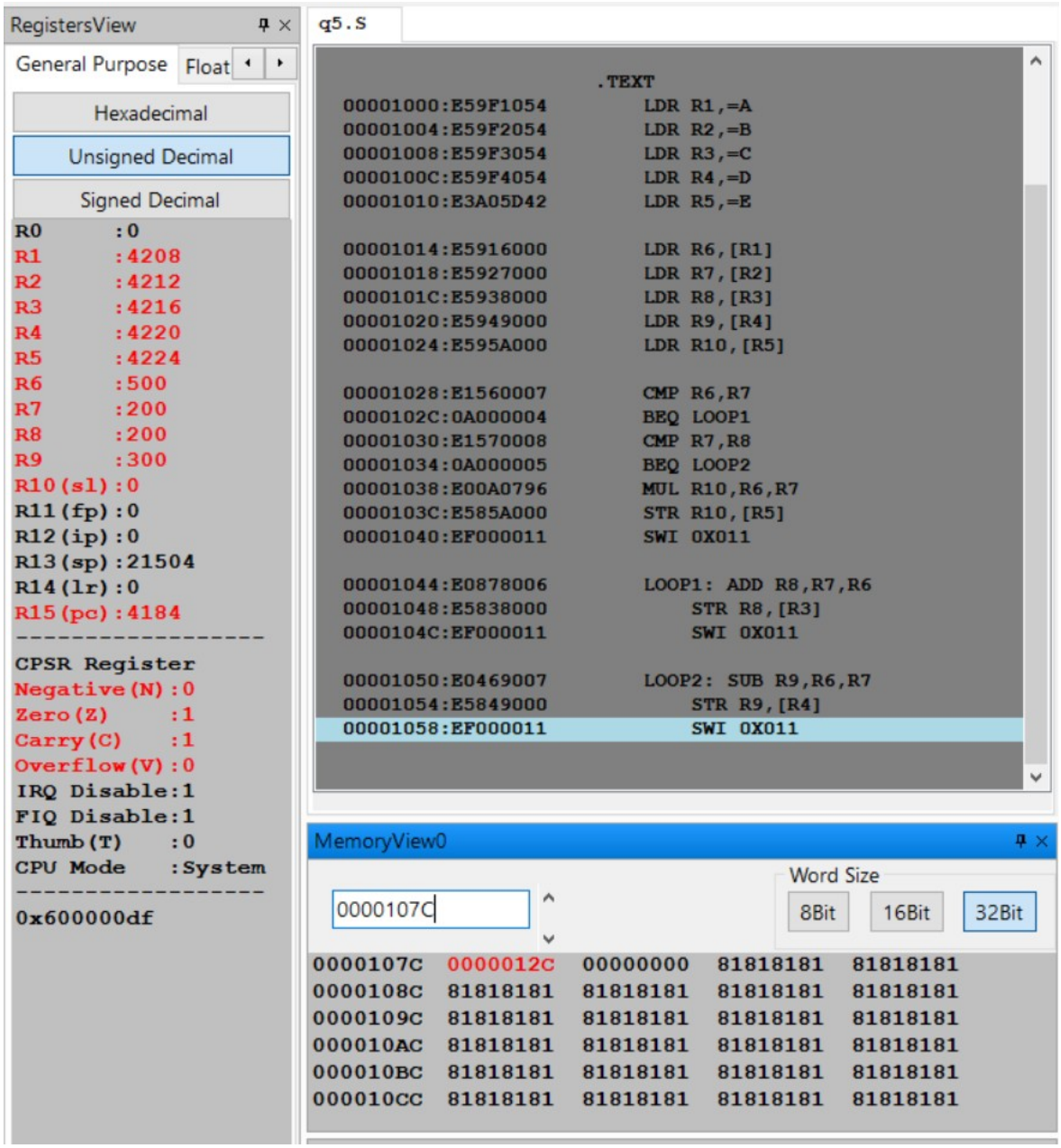
```
1   .DATA
2       A: .WORD 200
3       B: .WORD 200
4       C: .WORD 300
5       D: .WORD 0
6       E: .WORD 0
7
8   .TEXT
9       LDR R1,=A
10      LDR R2,=B
11      LDR R3,=C
12      LDR R4,=D
13      LDR R5,=E
14
15      LDR R6,[R1]
16      LDR R7,[R2]
17      LDR R8,[R3]
18      LDR R9,[R4]
19      LDR R10,[R5]
20
21      CMP R6,R7
22      BEQ LOOP1
23      CMP R7,R8
24      BEQ LOOP2
25      MUL R10,R6,R7
26      SWI 0X011
27
28      LOOP1: ADD R8,R7,R6
29          SWI 0X011
30
31      LOOP2: SUB R9,R7,R6
32          SWI 0X011
33
34
```

OUTPUT :

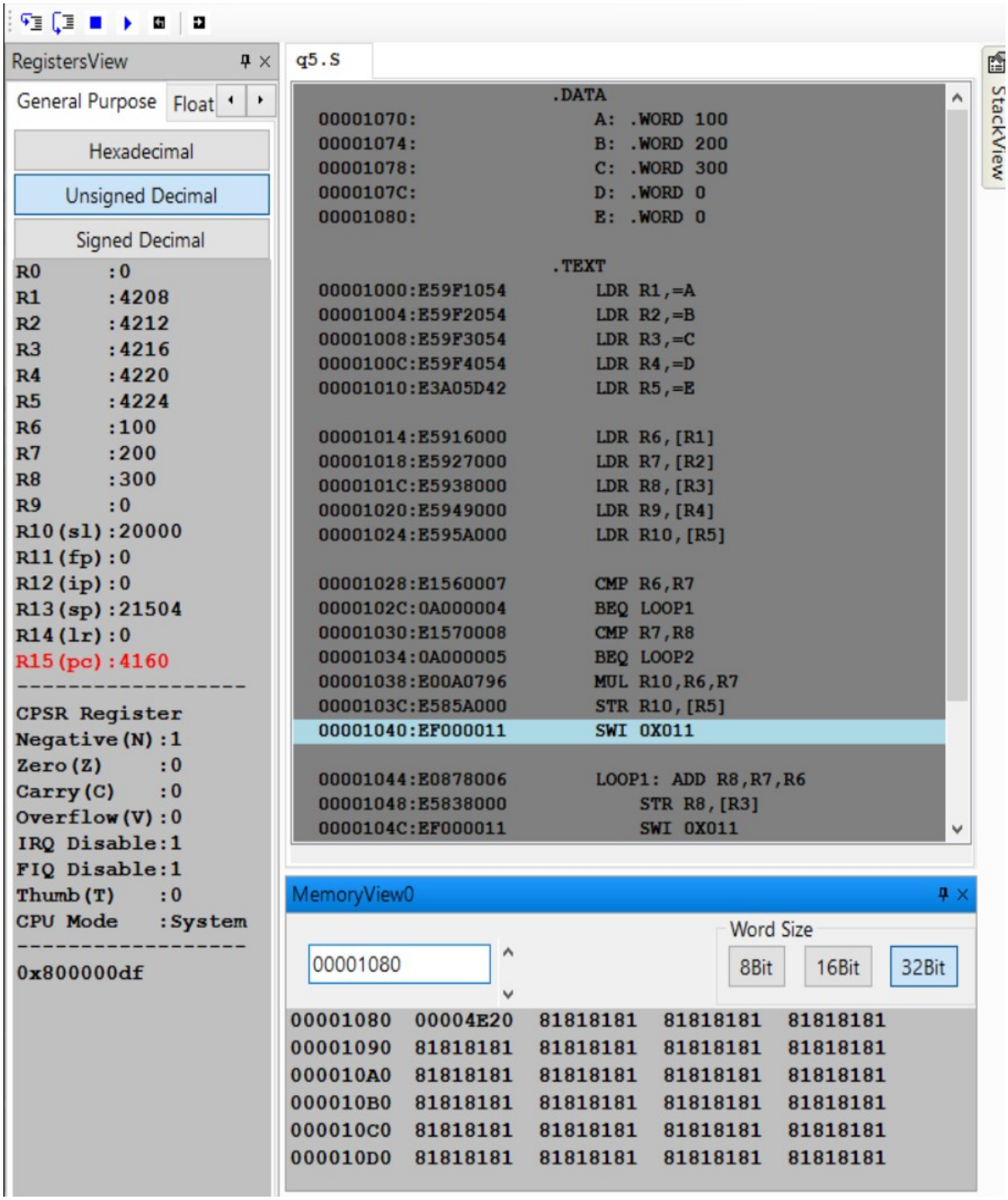
1.WHEN A = B



2. WHEN B=C

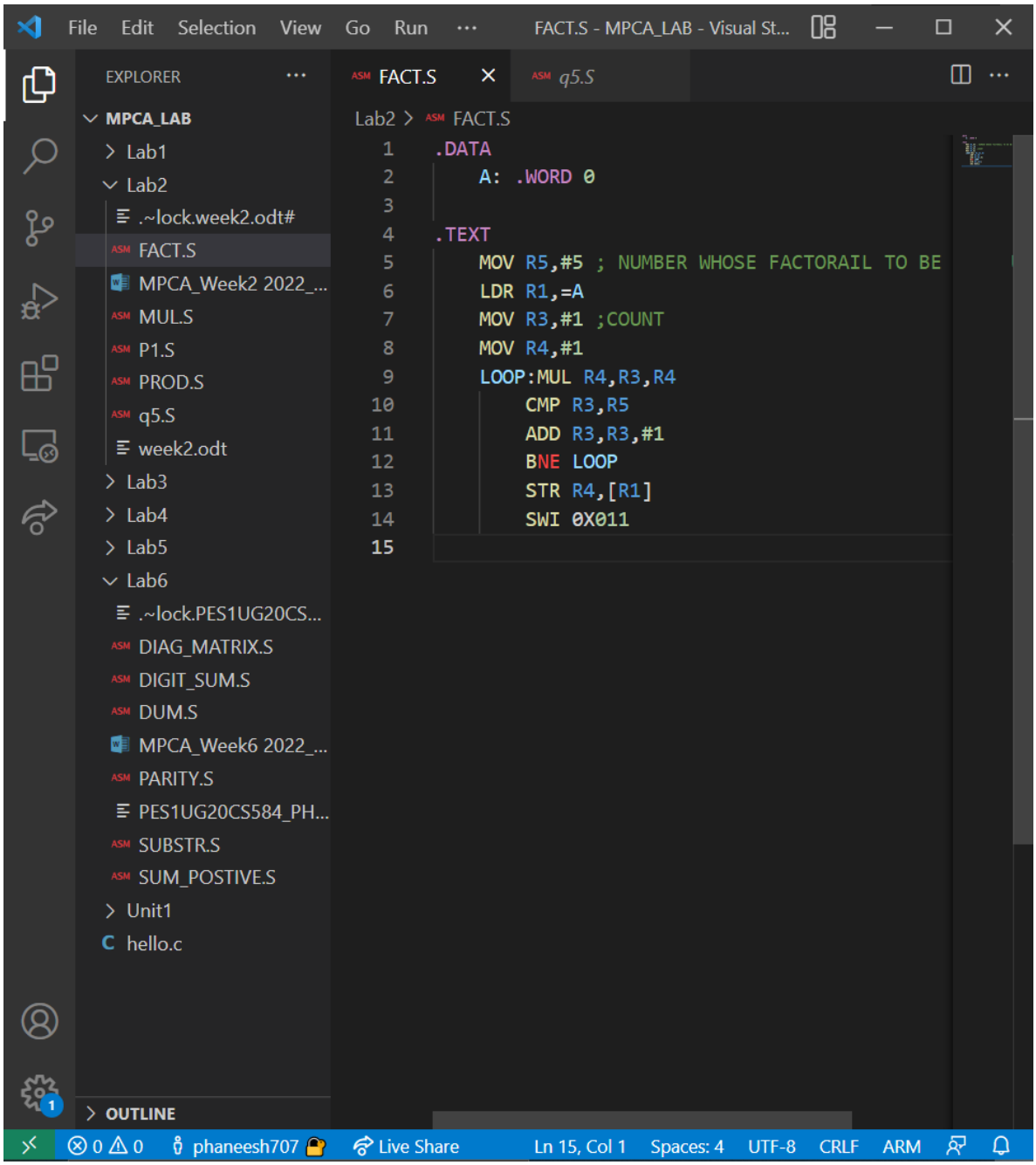

RegistersView
General Purpose
Float
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0
R1 :4208
R2 :4212
R3 :4216
R4 :4220
R5 :4224
R6 :500
R7 :200
R8 :200
R9 :300
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4184
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df
q5.S
.TEXT
00001000:E59F1054 LDR R1,=A
00001004:E59F2054 LDR R2,=B
00001008:E59F3054 LDR R3,=C
0000100C:E59F4054 LDR R4,=D
00001010:E3A05D42 LDR R5,=E
00001014:E5916000 LDR R6,[R1]
00001018:E5927000 LDR R7,[R2]
0000101C:E5938000 LDR R8,[R3]
00001020:E5949000 LDR R9,[R4]
00001024:E595A000 LDR R10,[R5]
00001028:E1560007 CMP R6,R7
0000102C:0A000004 BEQ LOOP1
00001030:E1570008 CMP R7,R8
00001034:0A000005 BEQ LOOP2
00001038:E00A0796 MUL R10,R6,R7
0000103C:E585A000 STR R10,[R5]
00001040:EF000011 SWI 0X011
00001044:E0878006 LOOP1: ADD R8,R7,R6
00001048:E5838000 STR R8,[R3]
0000104C:EF000011 SWI 0X011
00001050:E0469007 LOOP2: SUB R9,R6,R7
00001054:E5849000 STR R9,[R4]
00001058:EF000011 SWI 0X011
MemoryView0
0000107C
Word Size
8Bit
16Bit
32Bit
0000107C 0000012C 00000000 81818181 81818181
0000108C 81818181 81818181 81818181 81818181
0000109C 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181
000010BC 81818181 81818181 81818181 81818181
000010CC 81818181 81818181 81818181 81818181

3.IN ALL OTHER CASES

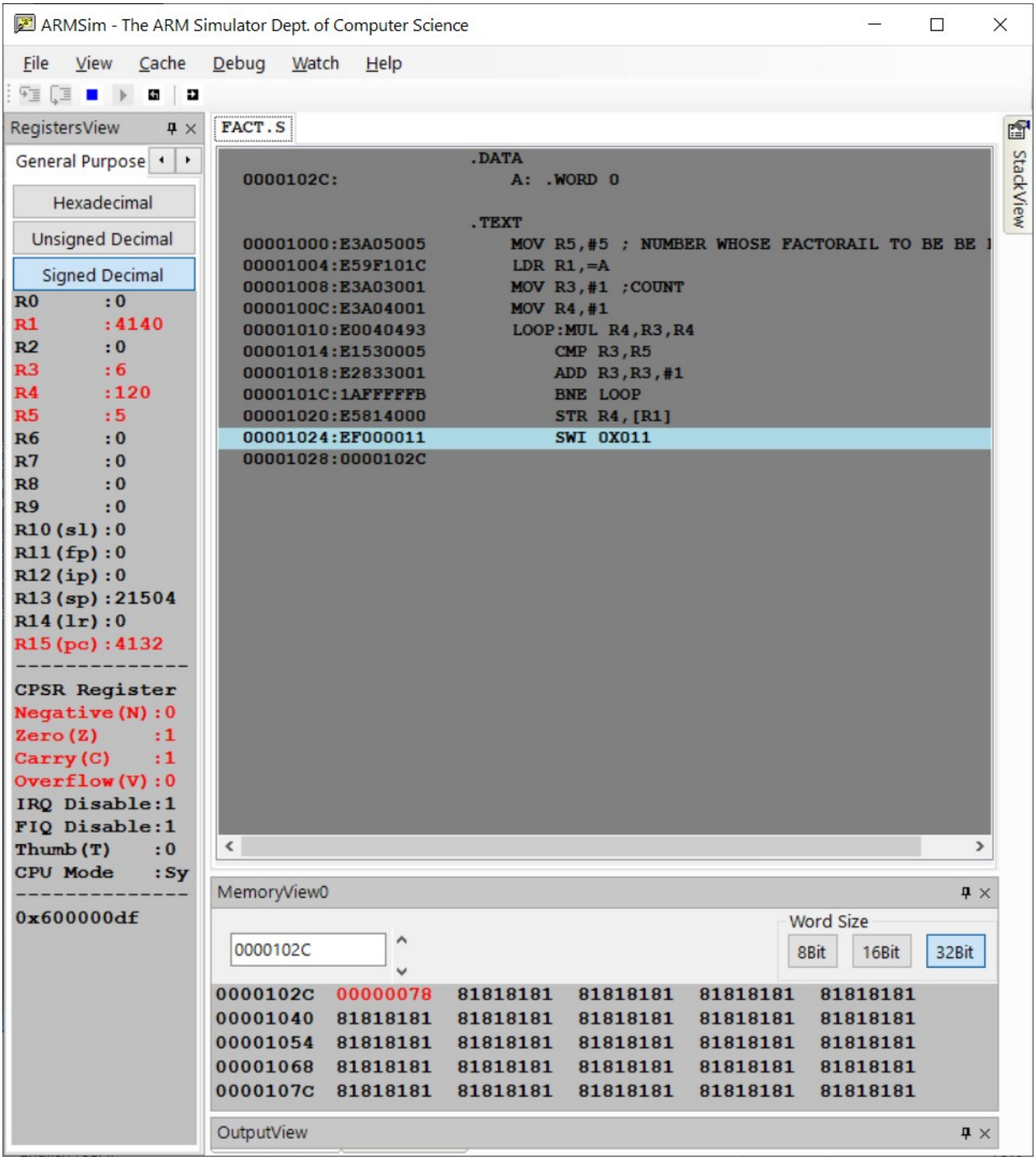


6. Write a program in ARM7TDMI-ISA to find the factorial of a number.

CODE:



OUTPUT:


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0
R1 :4140
R2 :0
R3 :6
R4 :120
R5 :5
R6 :0
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4132
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 :Sy
0x600000df
FACT.S
.DATA
0000102C: A: .WORD 0
.TEXT
00001000:E3A05005 MOV R5,#5 ; NUMBER WHOSE FACTORAIL TO BE BE
00001004:E59F101C LDR R1,=A
00001008:E3A03001 MOV R3,#1 ;COUNT
0000100C:E3A04001 MOV R4,#1
00001010:E0040493 LOOP:MUL R4,R3,R4
00001014:E1530005 CMP R3,R5
00001018:E2833001 ADD R3,R3,#1
0000101C:1AFFFFFB BNE LOOP
00001020:E5814000 STR R4,[R1]
00001024:EF000011 SWI 0X011
00001028:0000102C
StackView
MemoryView0
0000102C
Word Size
8Bit 16Bit 32Bit
0000102C 00000078 81818181 81818181 81818181 81818181
00001040 81818181 81818181 81818181 81818181 81818181
00001054 81818181 81818181 81818181 81818181 81818181
00001068 81818181 81818181 81818181 81818181 81818181
0000107C 81818181 81818181 81818181 81818181 81818181
OutputView