Department of Computer Science & Engineering

Microprocessor & Computer Architecture

MPCA-Laboratory/Assignment/Hands-on/Project

UE20CS252

NAME : VISHWAS M

SEC : F

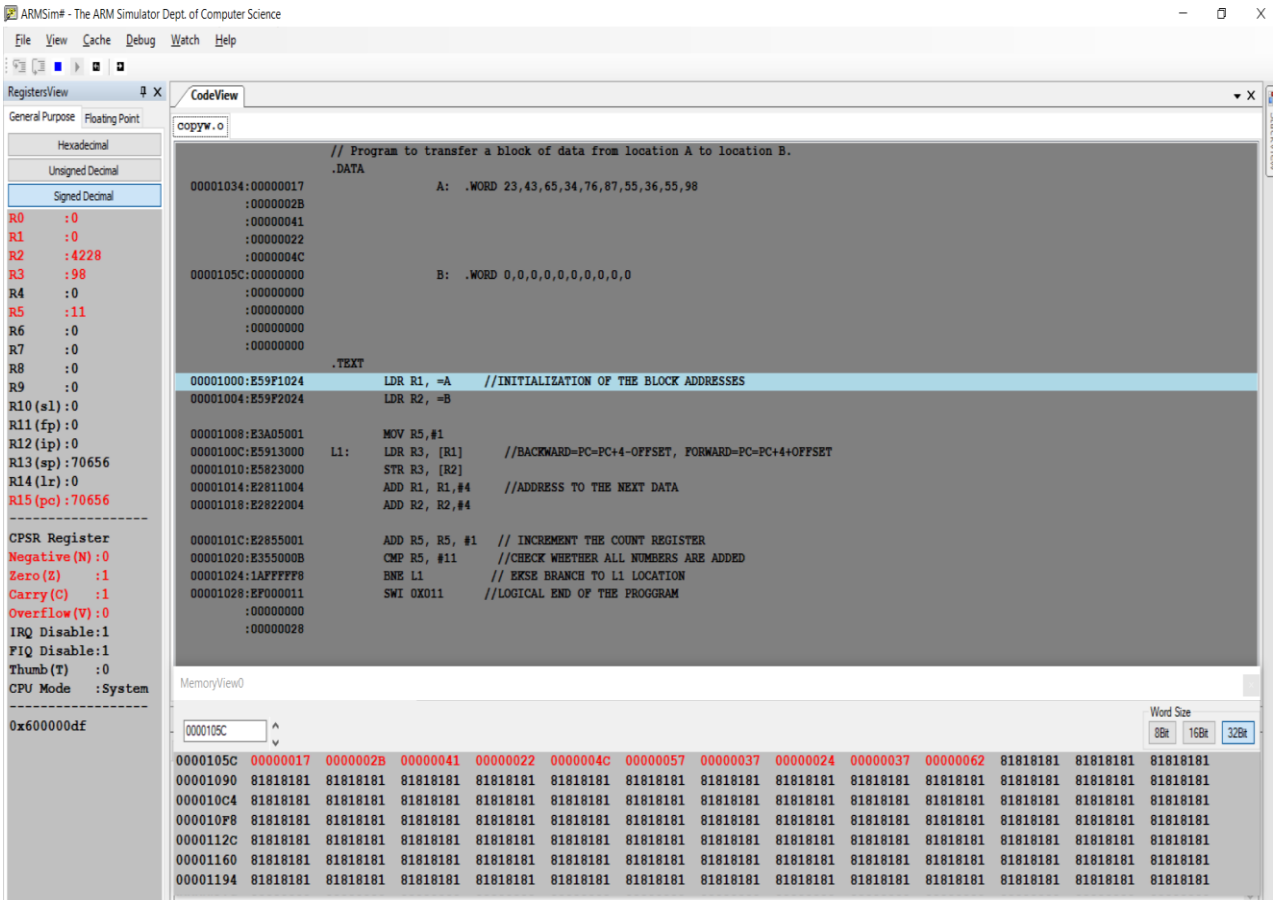
SRN : PES2UG20CS390

DATE : 31/01/2022

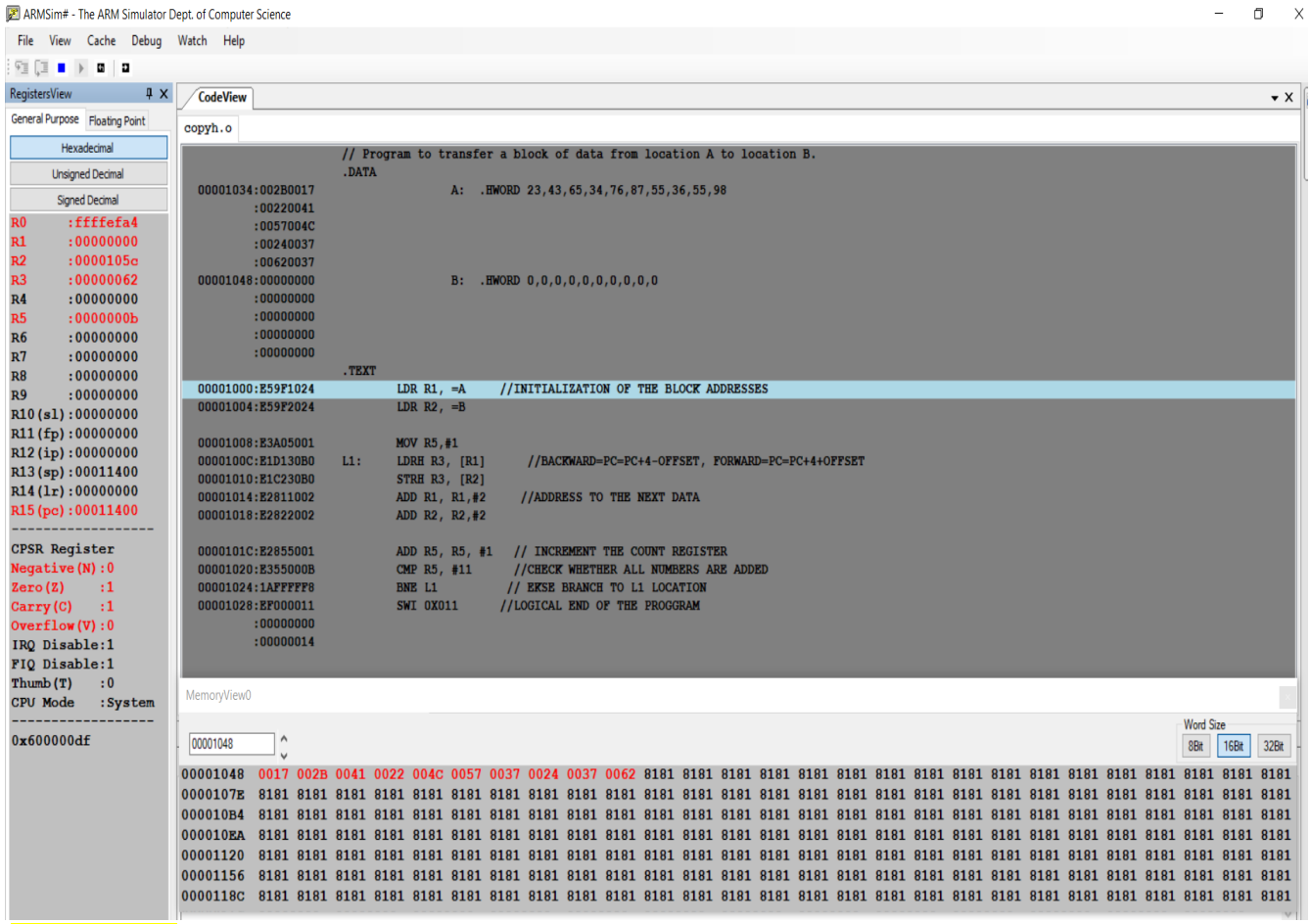
1. Write a program in ARM7TDMI-ISA to copy a block of N data items from Location A to Locatic B.
   a. Use Full word (.word directive)
   b. Use Half word(.Hword directive)
   c. Use Byte wise (.Byte directive)

Screenshots:

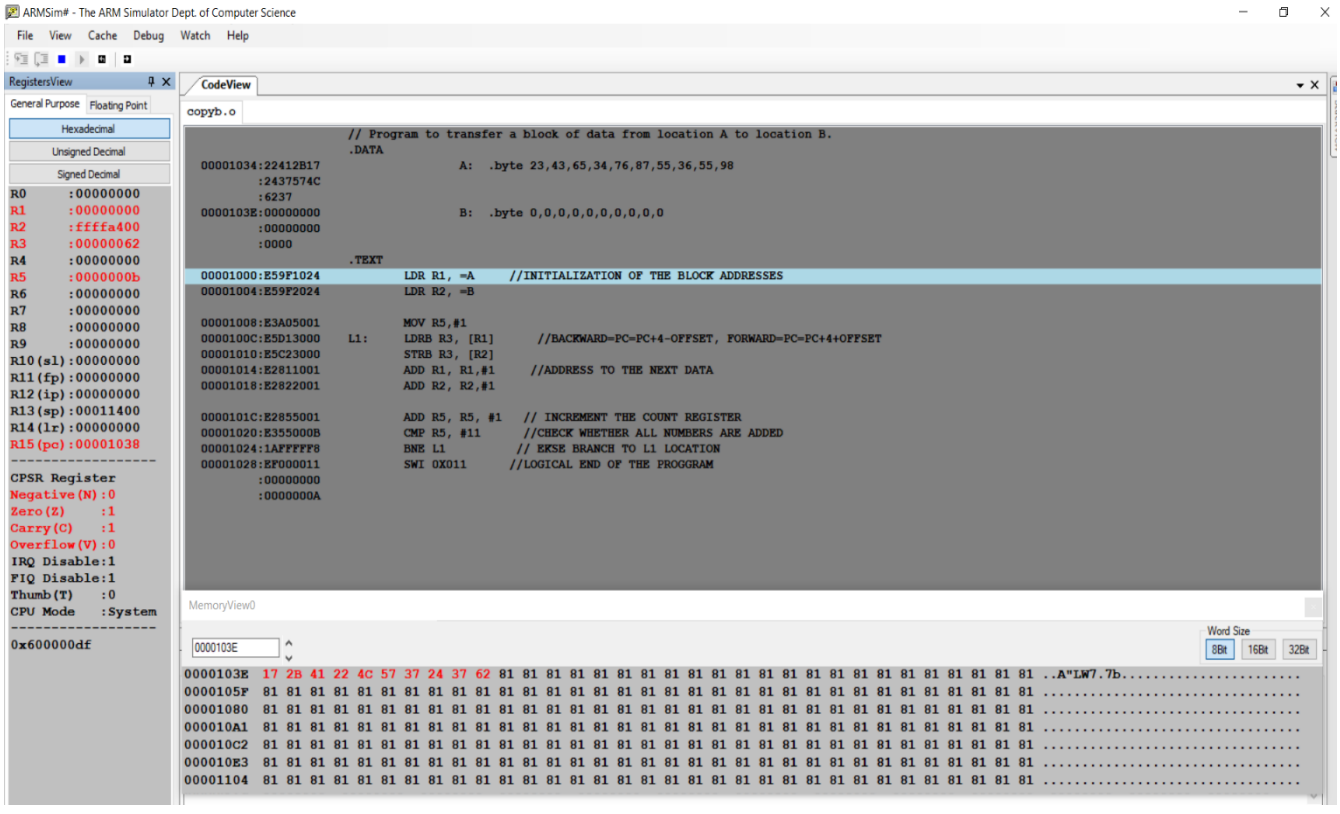
a) Using full word.



b) using half word.


ARMSim# - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :ffffefa4
R1 :00000000
R2 :0000105c
R3 :00000062
R4 :00000000
R5 :0000000b
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
CodeView
copyh.o
// Program to transfer a block of data from location A to location B.
.DATA
00001034:002B0017 A: .HWORD 23,43,65,34,76,87,55,36,55,98
:00220041
:0057004C
:00240037
:00620037
00001048:00000000 B: .HWORD 0,0,0,0,0,0,0,0,0,0
:00000000
:00000000
:00000000
:00000000
.TEXT
00001000:E59F1024 LDR R1, =A //INITIALIZATION OF THE BLOCK ADDRESSES
00001004:E59F2024 LDR R2, =B
00001008:E3A05001 MOV R5,#1
0000100C:E1D130B0 L1: LDRH R3, [R1] //BACKWARD=PC=PC+4-OFFSET, FORWARD=PC=PC+4+OFFSET
00001010:E1C230B0 STRH R3, [R2]
00001014:E2811002 ADD R1, R1,#2 //ADDRESS TO THE NEXT DATA
00001018:E2822002 ADD R2, R2,#2
0000101C:E2855001 ADD R5, R5, #1 // INCREMENT THE COUNT REGISTER
00001020:E355000B CMP R5, #11 //CHECK WHETHER ALL NUMBERS ARE ADDED
00001024:1AFFFFF8 BNE L1 // EKSE BRANCH TO L1 LOCATION
00001028:EF000011 SWI 0X011 //LOGICAL END OF THE PROGGRAM
:00000000
:00000014
MemoryView0
00001048
Word Size
8Bit 16Bit 32Bit
00001048 0017 002B 0041 0022 004C 0057 0037 0024 0037 0062 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000107E 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010B4 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010EA 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001120 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001156 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000118C 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181

c)**using byte.**


ARMSim# - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :ffffa400
R3 :00000062
R4 :00000000
R5 :0000000b
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):00001038
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
CodeView
copyb.o
// Program to transfer a block of data from location A to location B.
.DATA
00001034:22412B17 A: .byte 23,43,65,34,76,87,55,36,55,98
:2437574C
:6237
0000103E:00000000 B: .byte 0,0,0,0,0,0,0,0,0,0
:00000000
:0000
.TEXT
00001000:E59F1024 LDR R1, =A //INITIALIZATION OF THE BLOCK ADDRESSES
00001004:E59F2024 LDR R2, =B
00001008:E3A05001 MOV R5,#1
0000100C:E5D13000 L1: LDRB R3, [R1] //BACKWARD=PC=PC+4-OFFSET, FORWARD=PC=PC+4+OFFSET
00001010:E5C23000 STRB R3, [R2]
00001014:E2811001 ADD R1, R1,#1 //ADDRESS TO THE NEXT DATA
00001018:E2822001 ADD R2, R2,#1
0000101C:E2855001 ADD R5, R5, #1 // INCREMENT THE COUNT REGISTER
00001020:E355000B CMP R5, #11 //CHECK WHETHER ALL NUMBERS ARE ADDED
00001024:1AFFFFF8 BNE L1 // EKSE BRANCH TO L1 LOCATION
00001028:EF000011 SWI 0X011 //LOGICAL END OF THE PROGGRAM
:00000000
:0000000A
MemoryView0
0000103E
Word Size
8Bit 16Bit 32Bit
0000103E 17 2B 41 22 4C 57 37 24 37 62 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 ..A"LW7.7b.....
0000105F 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 .....
00001080 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 .....
000010A1 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 .....
000010C2 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 .....
000010E3 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 .....
00001104 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 .....

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)

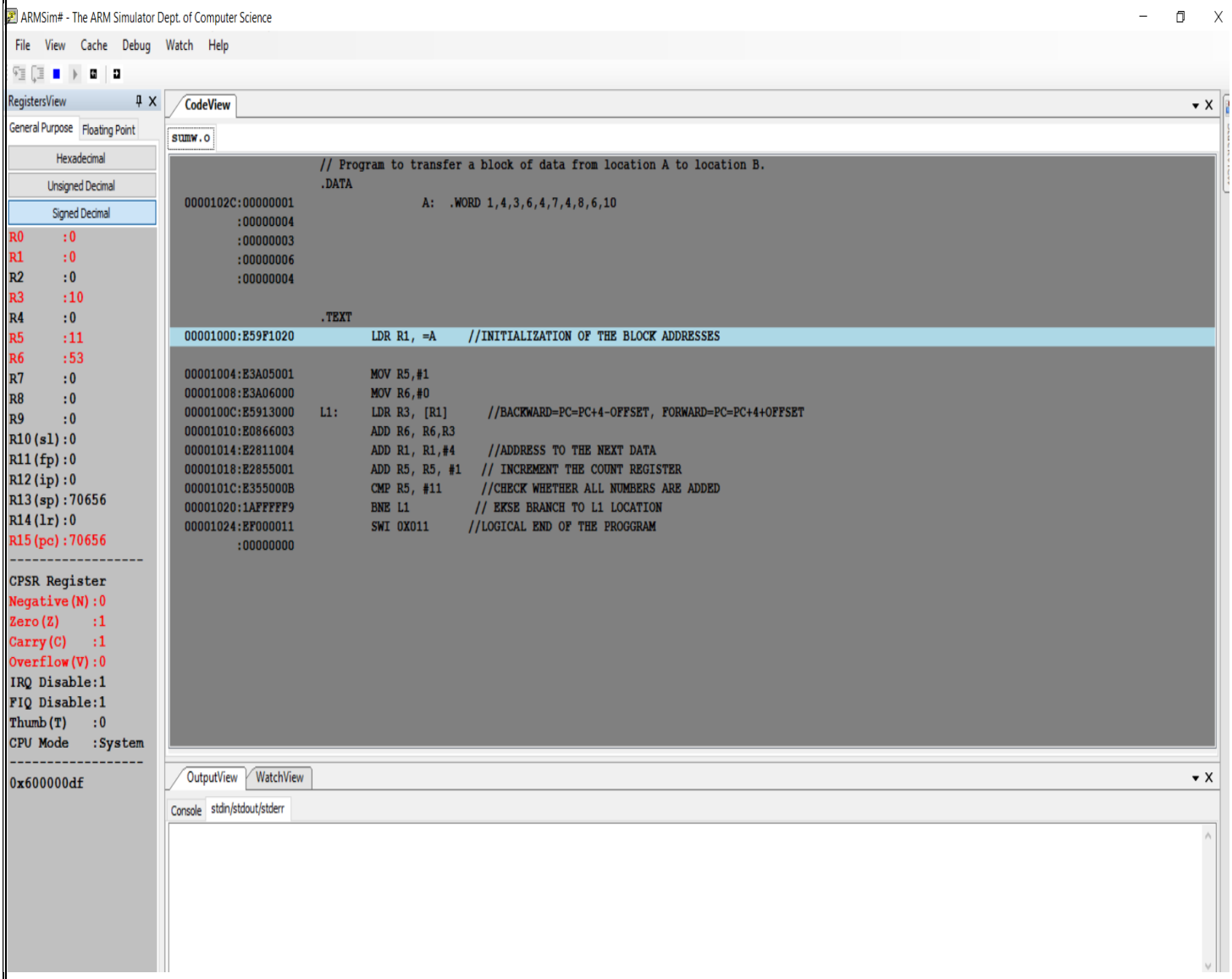
b. Use Half word(.Hword directive)

c. Use Byte wise (.Byte directive)

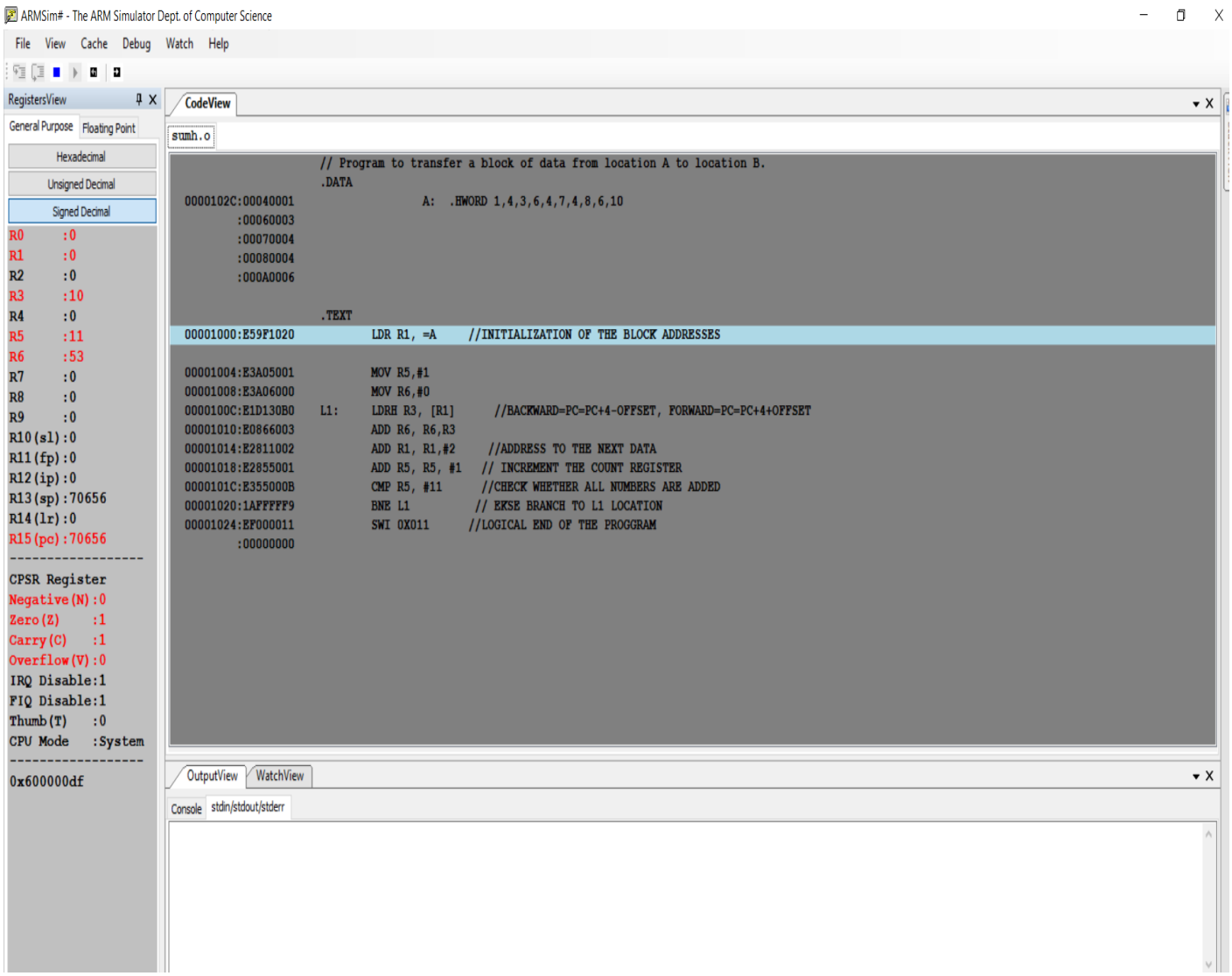
{1,2,3,4,5}

Screenshots:

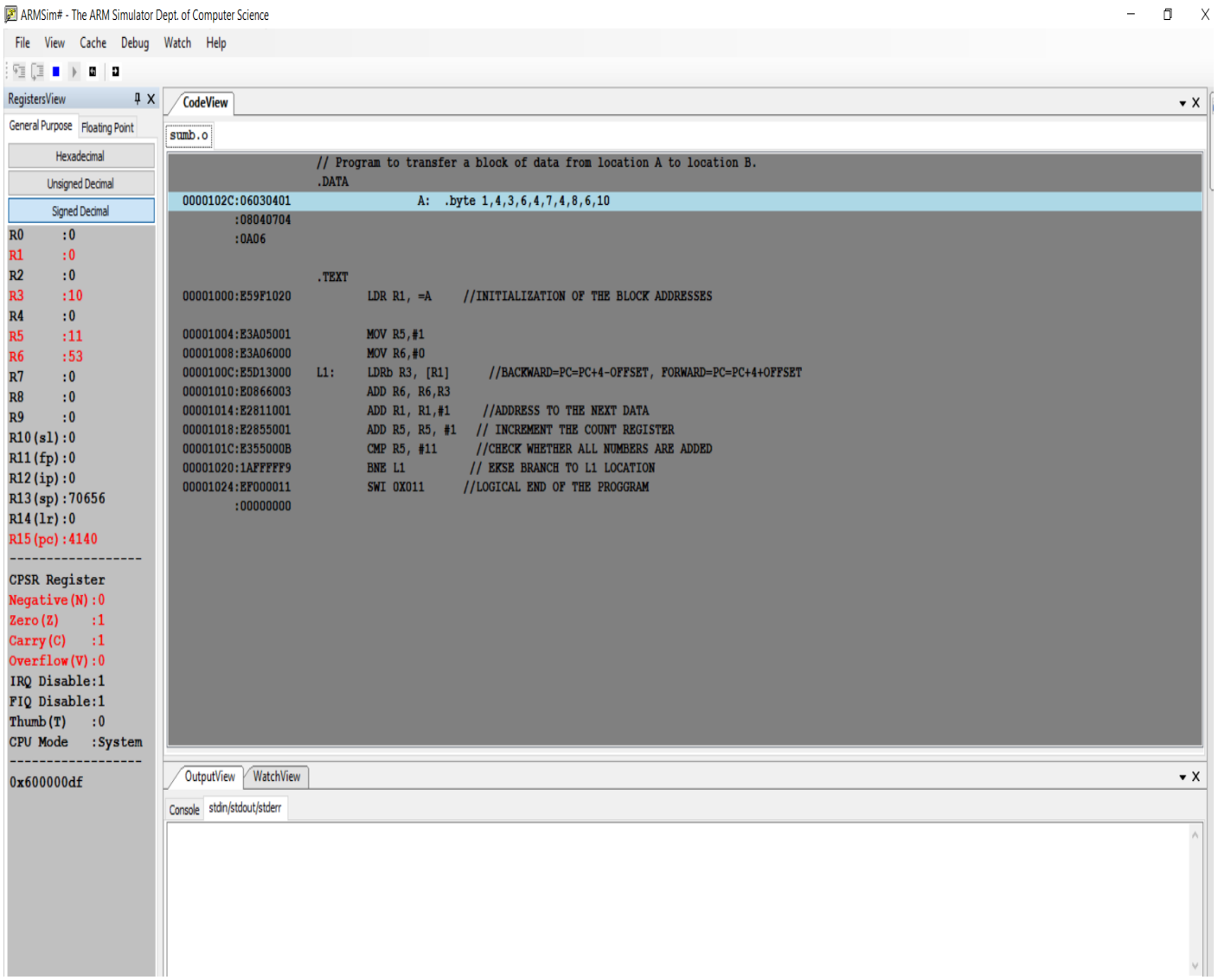
a) using full word.



b)Using half word.

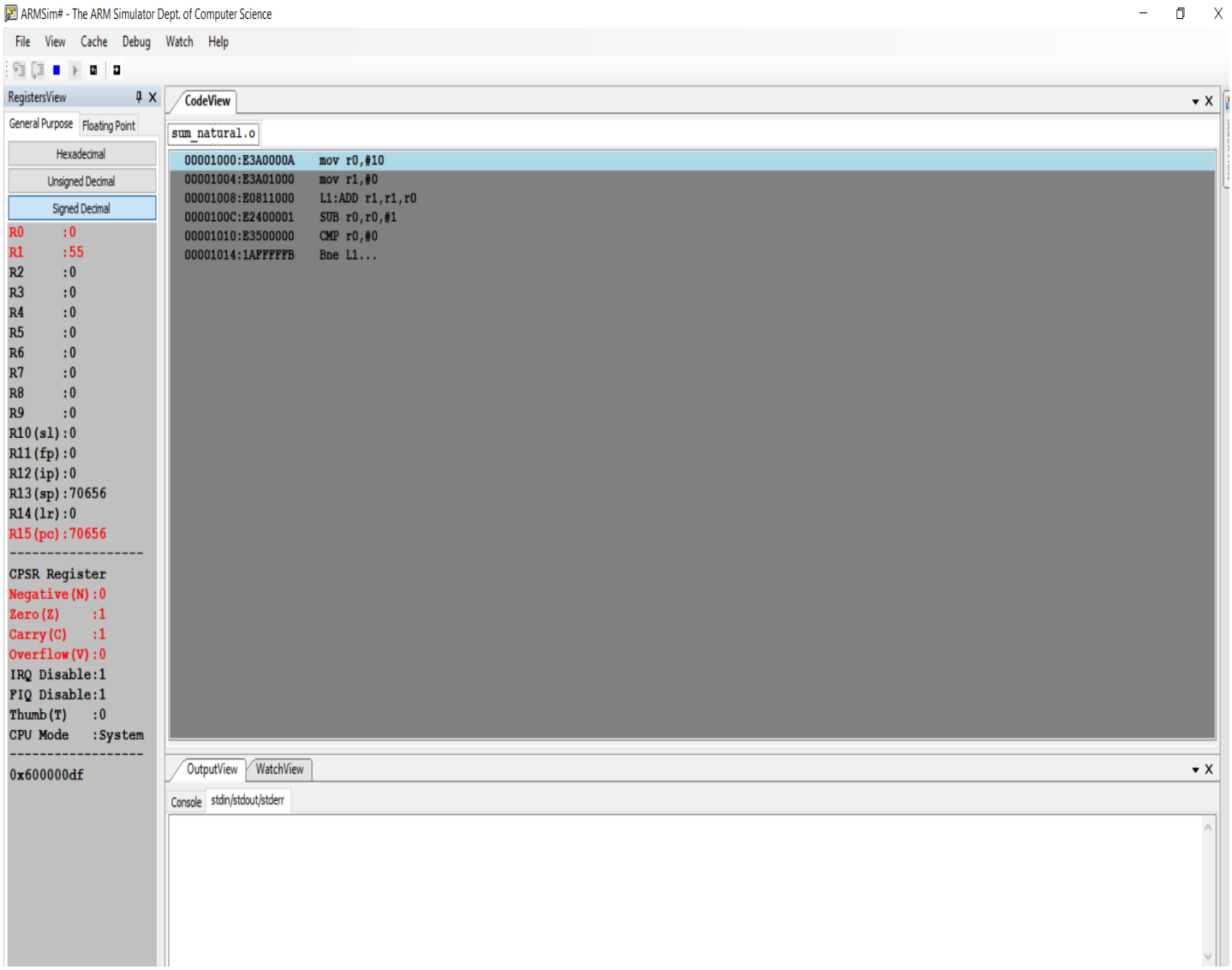


c)Using byte.



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

Screenshot:



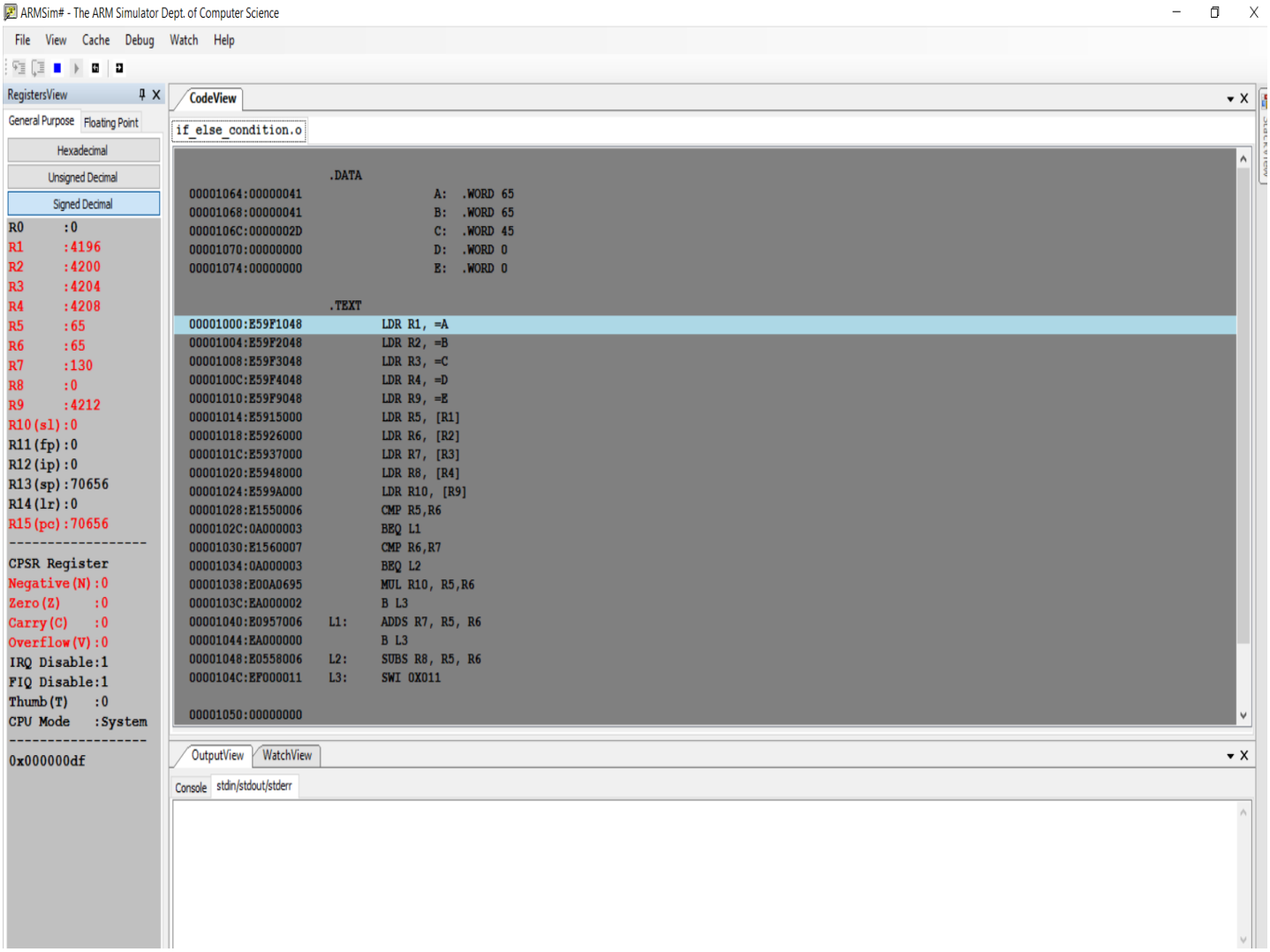
5. Convert the following statement in C language into an ALP using ARM7TDMI – ISA.
   IF([A]==[B]) then C=[A]+[B];

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

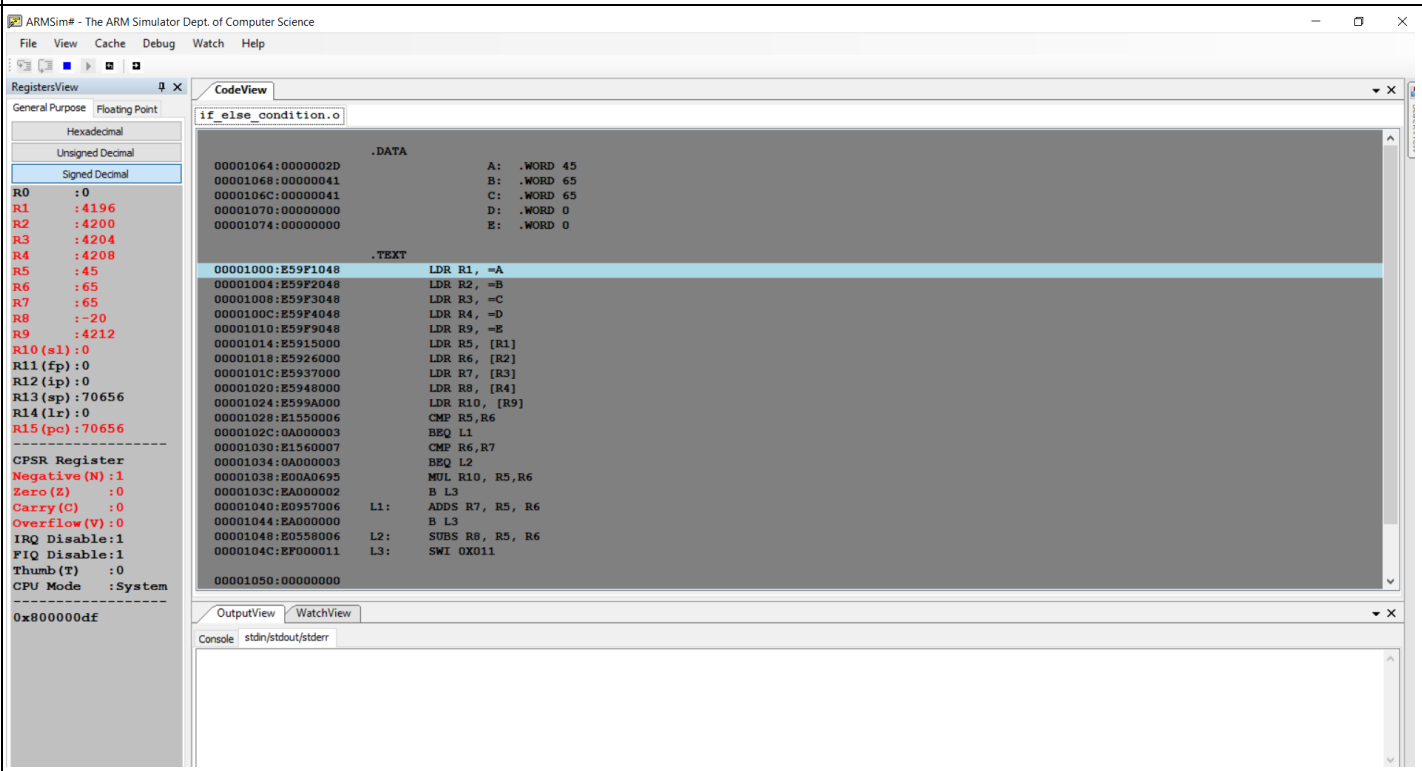
Where A,B C, D & E are memory locations.

Screenshots:

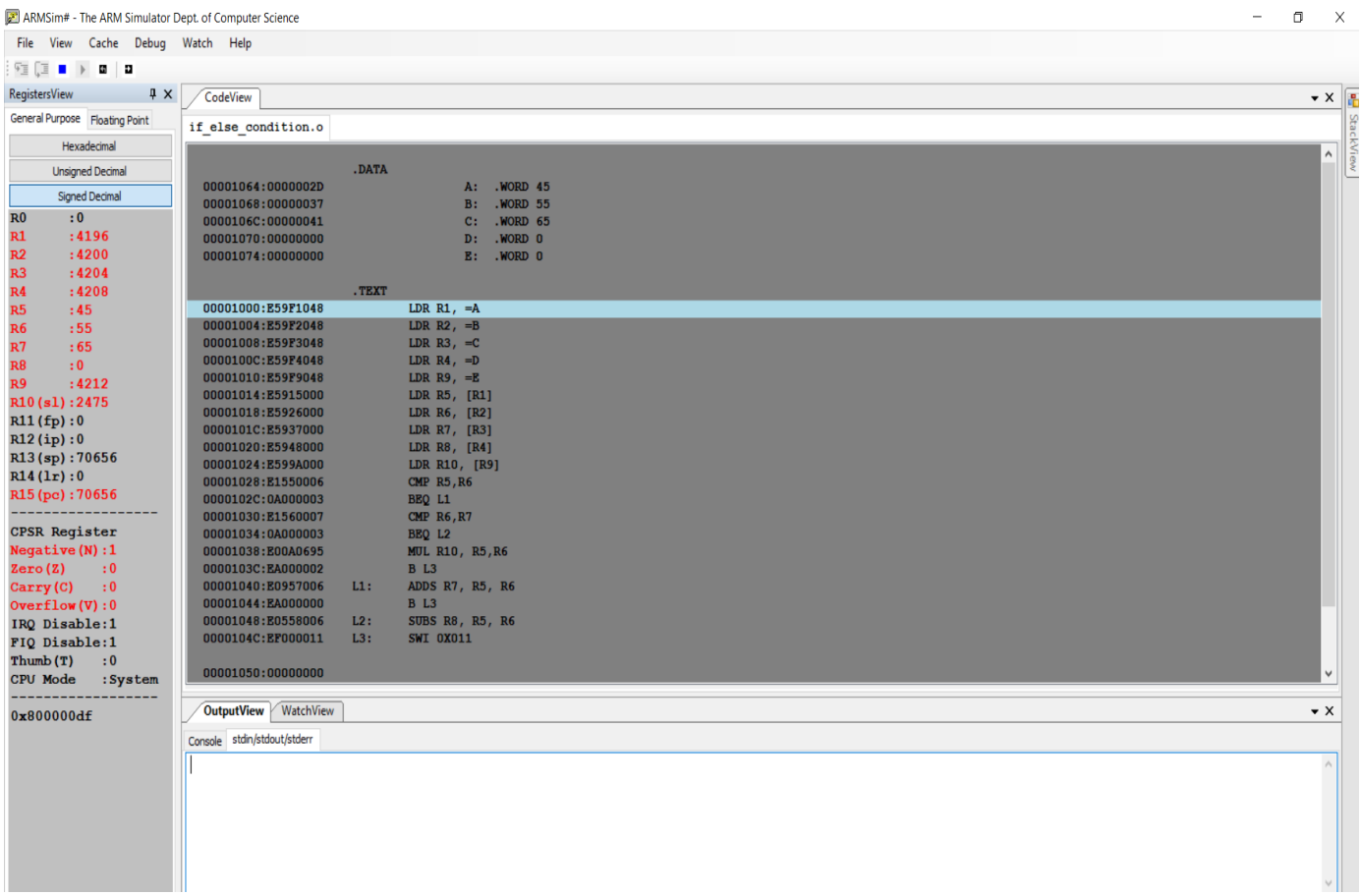
a)when A and B are equal:



b) when B and C are equal:

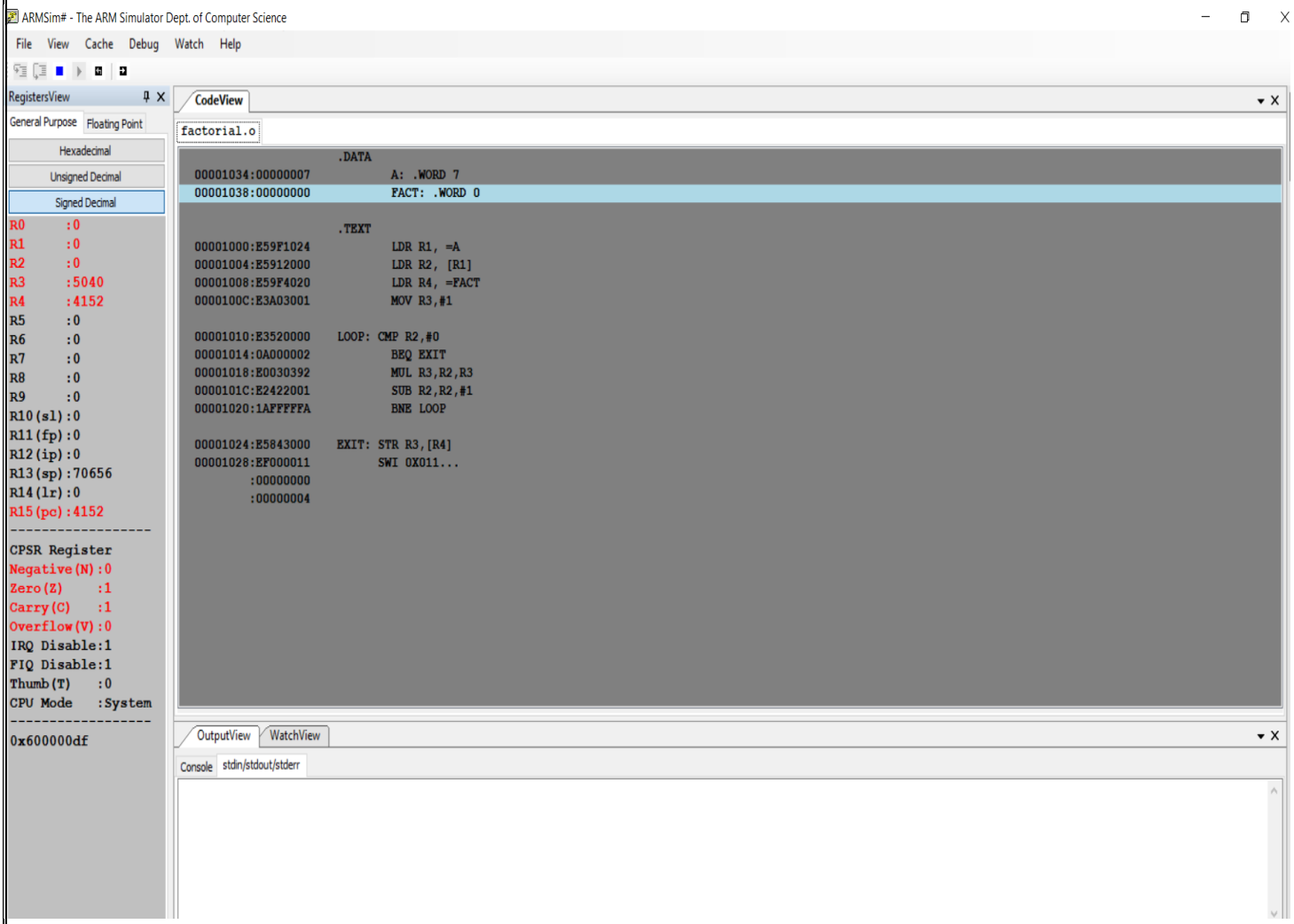


c) When A,B,C are different:



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

Screenshot:



Execute question 1,2,3,5,6