Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name:Nihal T M	SRN:PES2UG21CS333	Section:F

Week#____2_____ Program Number: ____1___

Title of the Program

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)

b. Use Half word(.hword directive)

c. Use Byte wise (.Byte directive)

I. ARM Assembly Code

Code:

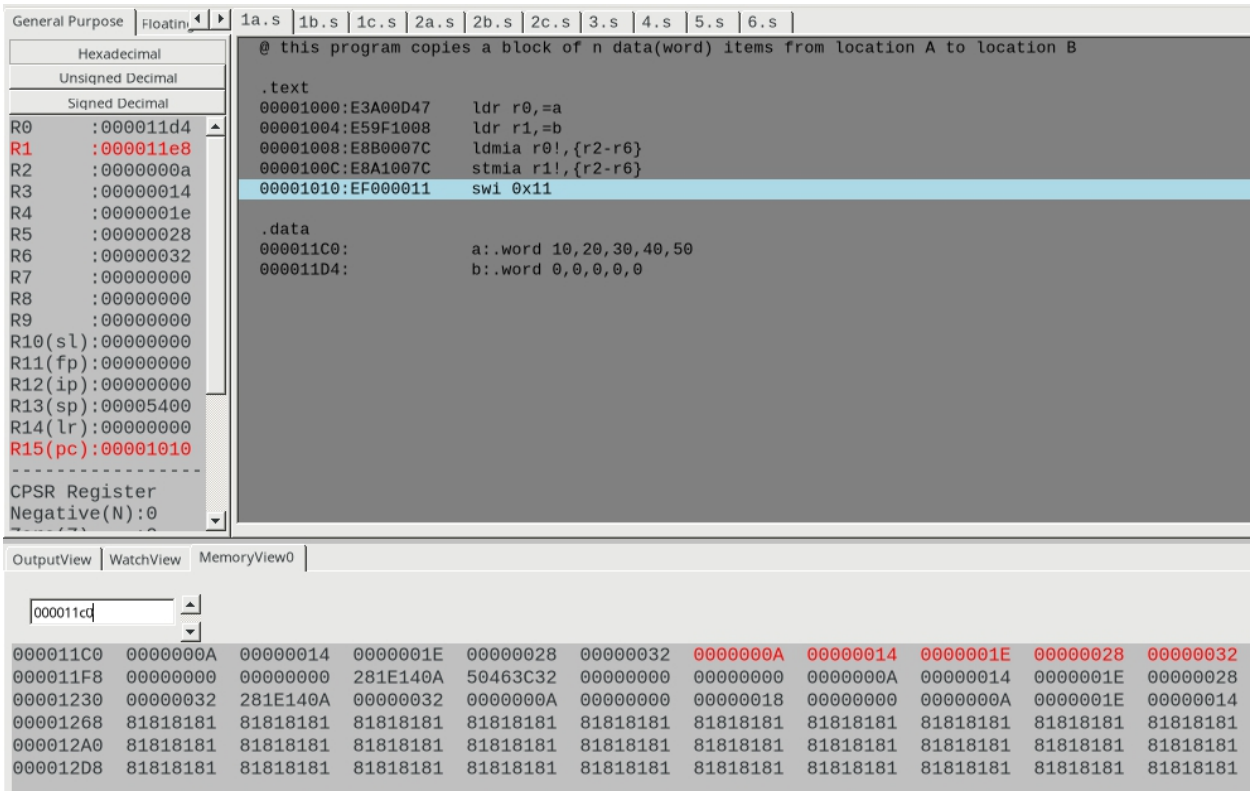
```
@ this program copies a block of n data(word) items from
location A to location B

.text
    ldr r0,=a
    ldr r1,=b
    ldmia r0!,{r2-r6}
```

```
stmia r1!,{r2-r6}
swi 0x11

.data
a:.word 10,20,30,40,50
b:.word 0,0,0,0,0
```

II. Output Screen Shots (Three)

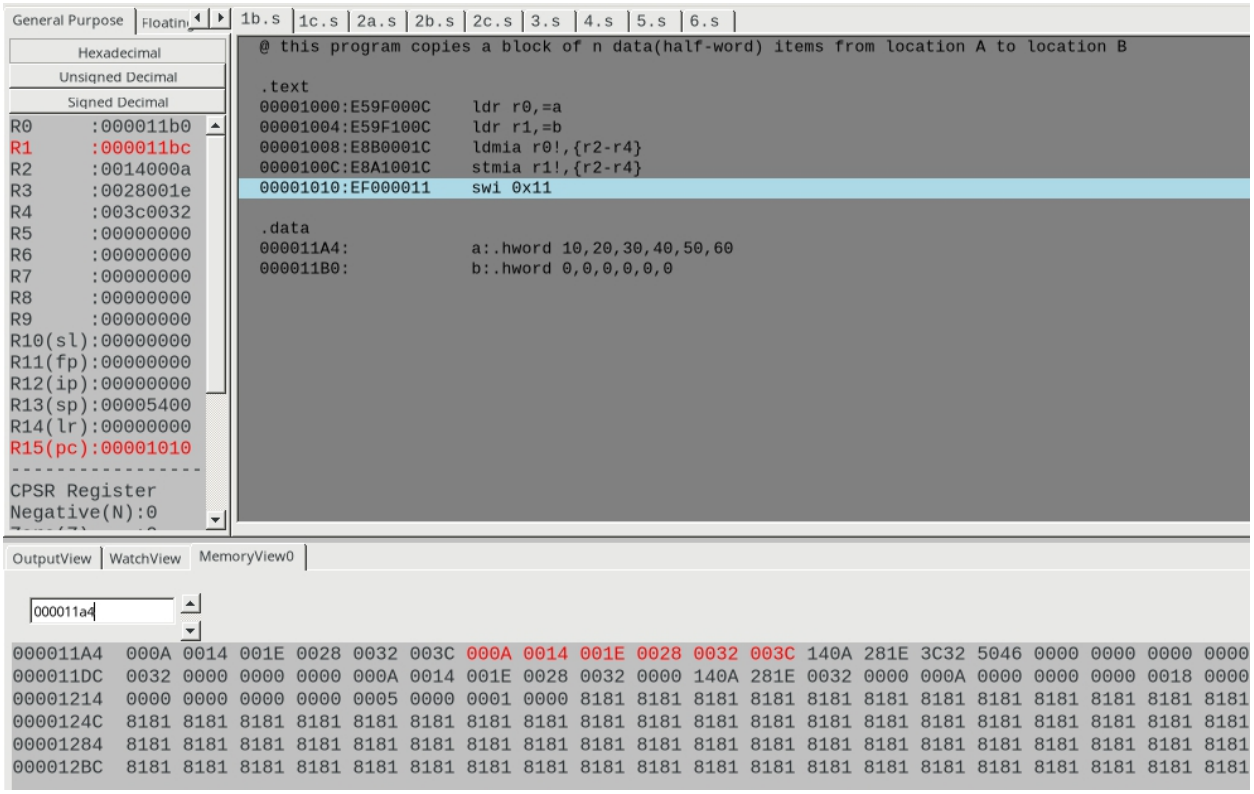
The output should be verified for word, half word, byte

i. word:



ii. half-word
code:
@ this program copies a block of n data(half-word) items from location A to location B

```
.text
        ldr r0,=a
        ldr r1,=b
        ldmia r0!,{r2-r4}
        stmia r1!,{r2-r4}
        swi 0x11

.data
        a:.hword 10,20,30,40,50,60
        b:.hword 0,0,0,0,0,0
```

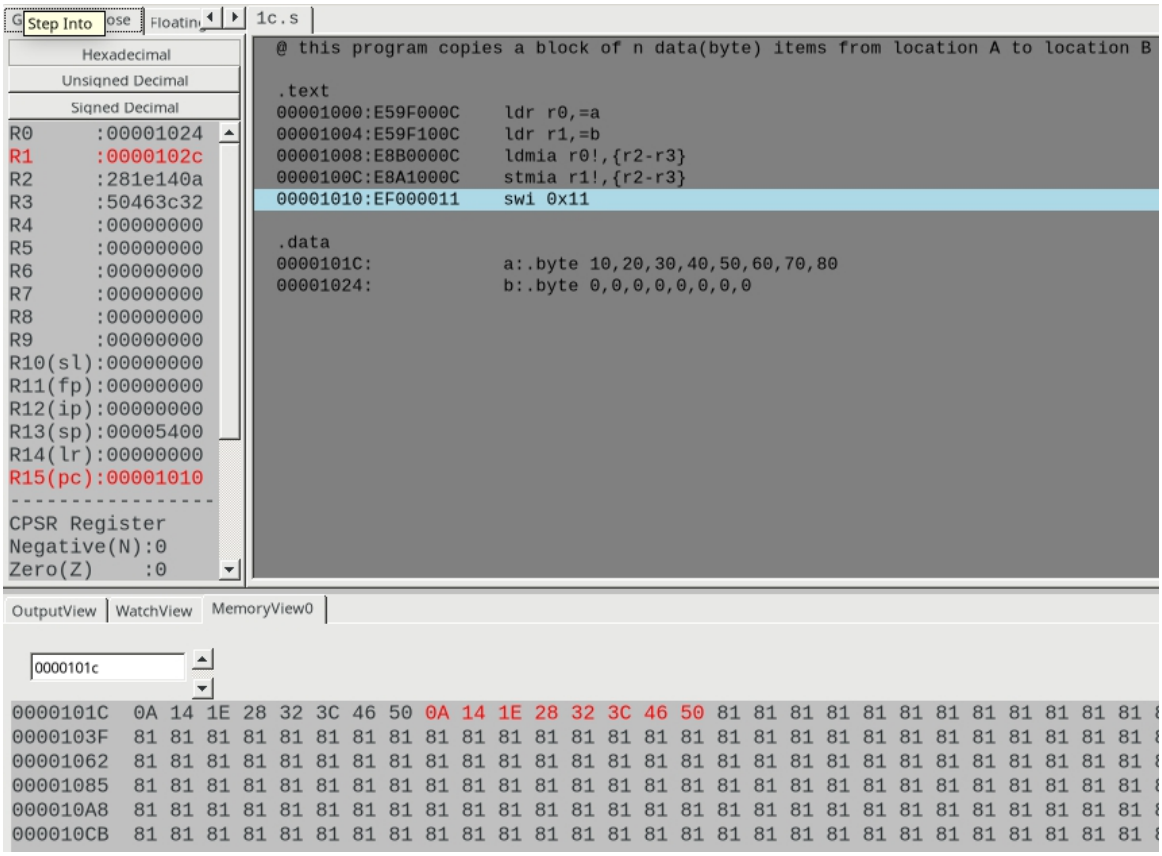


iii. byte

code:

@ this program copies a block of n data(byte) items from location A to location B

```
.text
        ldr r0,=a
        ldr r1,=b
        ldmia r0!,{r2-r3}
        stmia r1!,{r2-r3}
        swi 0x11
.data
        a:.byte 10,20,30,40,50,60,70,80
        b:.byte 0,0,0,0,0,0,0,0
```



Week#____2_____ Program Number: ____2___

Title of the Program

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)

I.ARM Assembly Code

Code:

i. word:

```
@ find sum of n words
.text
        ldr r0,=a
        ldr r1,=sum
        mov r2,#5
        mov r3,#0
        loop:
                ldr r4,[r0]
                add r3,r3,r4
                add r0,r0,#4
                subs r2,r2,#1
```

```
            bne loop
            str r3,[r1]
.data
    a:.word 10,20,30,40,50
    sum:.word 0
```

ii. half-word:

```
@ find sum of n half-words
.text
    ldr r0,=a
    ldr r1,=sum
    mov r2,#5
    mov r3,#0
    loop:
        ldrh r4,[r0]
        add r3,r3,r4
        add r0,r0,#2
        subs r2,r2,#1
        bne loop
    strh r3,[r1]
.data
    a:.hword 10,20,30,40,50
    sum:.hword 0
```

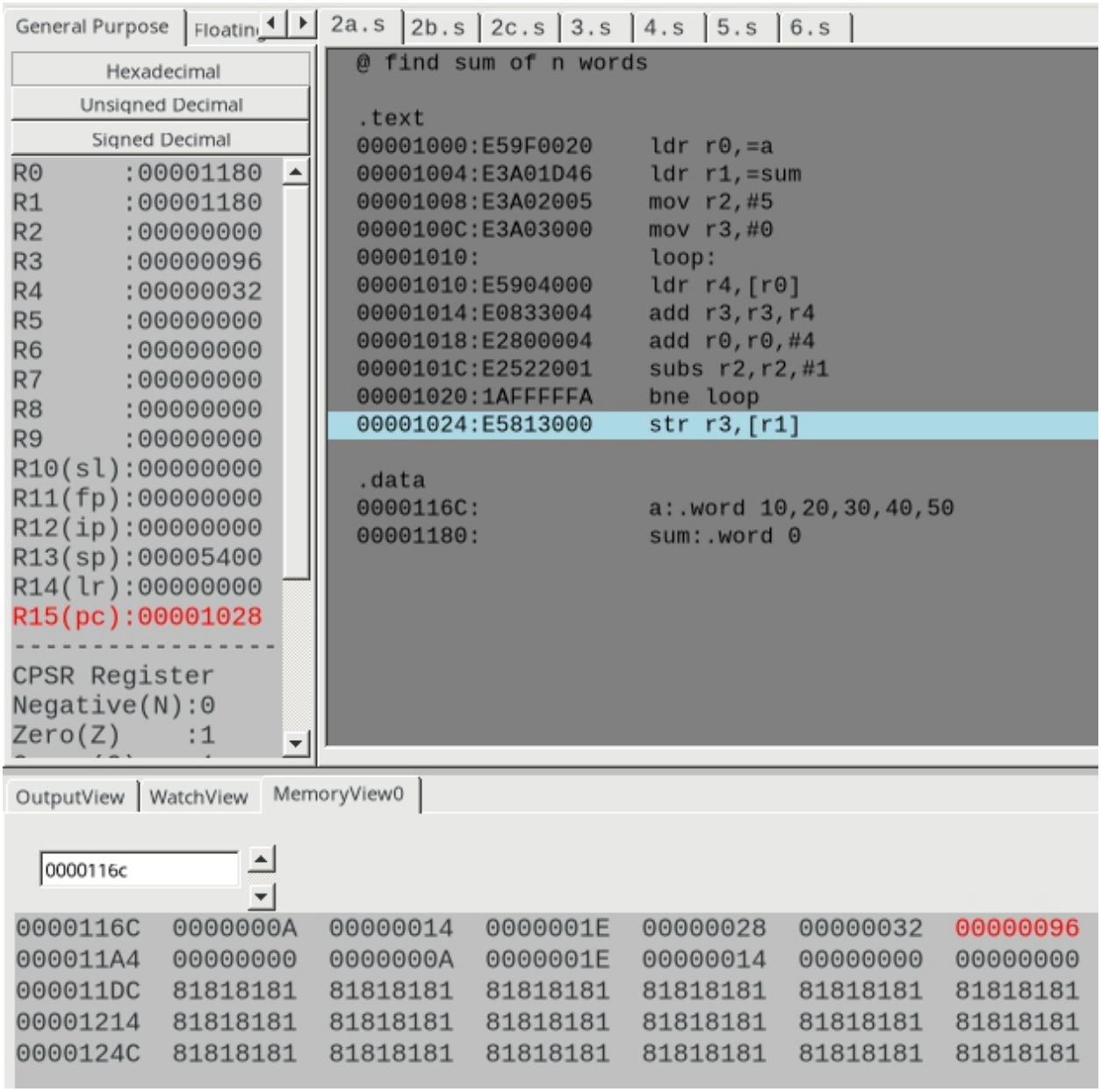
iii. byte:

```
@ find sum of n bytes
.text
	ldr r0,=a
	ldr r1,=sum
	mov r2,#5
	mov r3,#0
	loop:
		ldrb r4,[r0]
		add r3,r3,r4
		add r0,r0,#1
		subs r2,r2,#1
		bne loop
	strb r3,[r1]

.data
	a:.byte 10,20,30,40,50
	sum:.byte 0
```

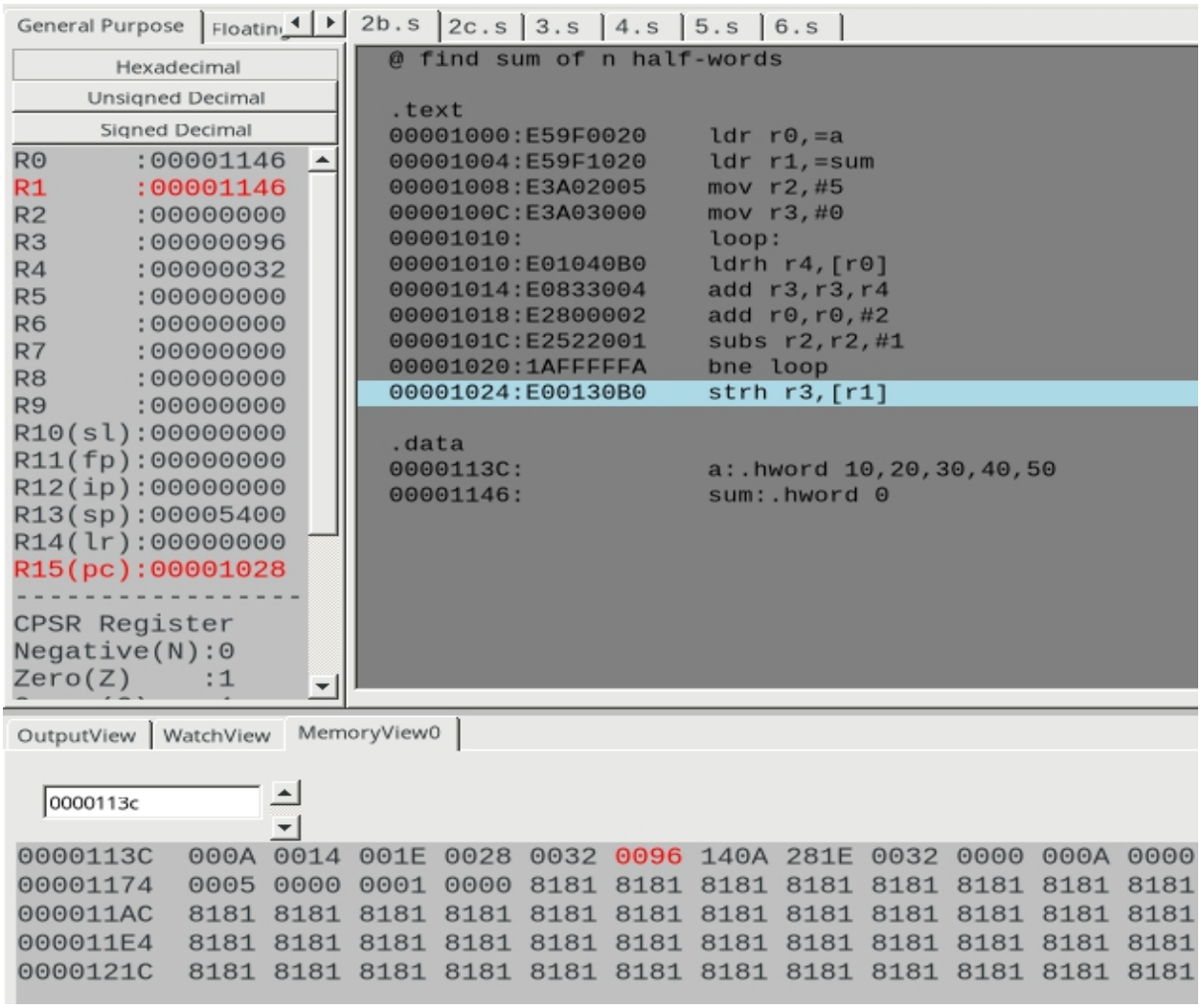
II. Output Screen Shots (Three)

The output should be verified for word, half word, byte

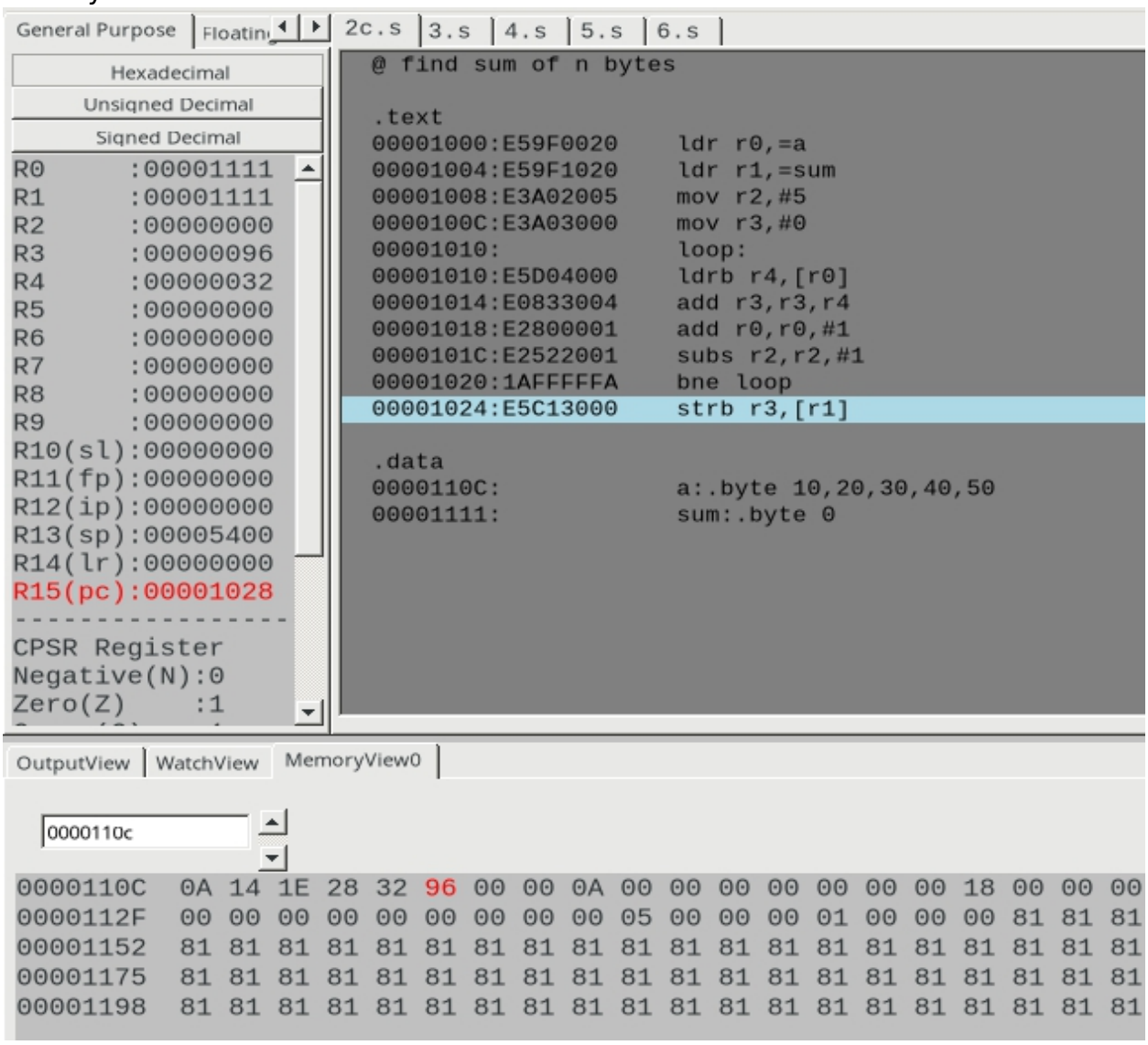
i. word:



ii. half-word:


General Purpose
Floatin
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00001146
R1 :00001146
R2 :00000000
R3 :00000096
R4 :00000032
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001028
CPSR Register
Negative(N):0
Zero(Z) :1
2b.s
2c.s
3.s
4.s
5.s
6.s
@ find sum of n half-words
.text
00001000:E59F0020 ldr r0,=a
00001004:E59F1020 ldr r1,=sum
00001008:E3A02005 mov r2,#5
0000100C:E3A03000 mov r3,#0
00001010: loop:
00001010:E01040B0 ldrh r4,[r0]
00001014:E0833004 add r3,r3,r4
00001018:E2800002 add r0,r0,#2
0000101C:E2522001 subs r2,r2,#1
00001020:1AFFFFFA bne loop
00001024:E00130B0 strh r3,[r1]
.data
0000113C: a:.hword 10,20,30,40,50
00001146: sum:.hword 0
OutputView
WatchView
MemoryView0
0000113c
0000113C 000A 0014 001E 0028 0032 0096 140A 281E 0032 0000 000A 0000
00001174 0005 0000 0001 0000 8181 8181 8181 8181 8181 8181 8181 8181
000011AC 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000011E4 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000121C 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181

iii. byte:


General Purpose
Floating
2c.s
3.s
4.s
5.s
6.s
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00001111
R1 :00001111
R2 :00000000
R3 :00000096
R4 :00000032
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001028
CPSR Register
Negative(N):0
Zero(Z) :1
@ find sum of n bytes
.text
00001000:E59F0020 ldr r0,=a
00001004:E59F1020 ldr r1,=sum
00001008:E3A02005 mov r2,#5
0000100C:E3A03000 mov r3,#0
00001010: loop:
00001010:E5D04000 ldrb r4,[r0]
00001014:E0833004 add r3,r3,r4
00001018:E2800001 add r0,r0,#1
0000101C:E2522001 subs r2,r2,#1
00001020:1AFFFFFA bne loop
00001024:E5C13000 strb r3,[r1]
.data
0000110C: a:.byte 10,20,30,40,50
00001111: sum:.byte 0
OutputView
WatchView
MemoryView0
0000110c
0000110C 0A 14 1E 28 32 96 00 00 0A 00 00 00 00 00 00 00 18 00 00 00
0000112F 00 00 00 00 00 00 00 00 00 05 00 00 00 01 00 00 00 81 81 81
00001152 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81
00001175 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81
00001198 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81

Week#____2_____ Program Number: ____3___

Title of the Program

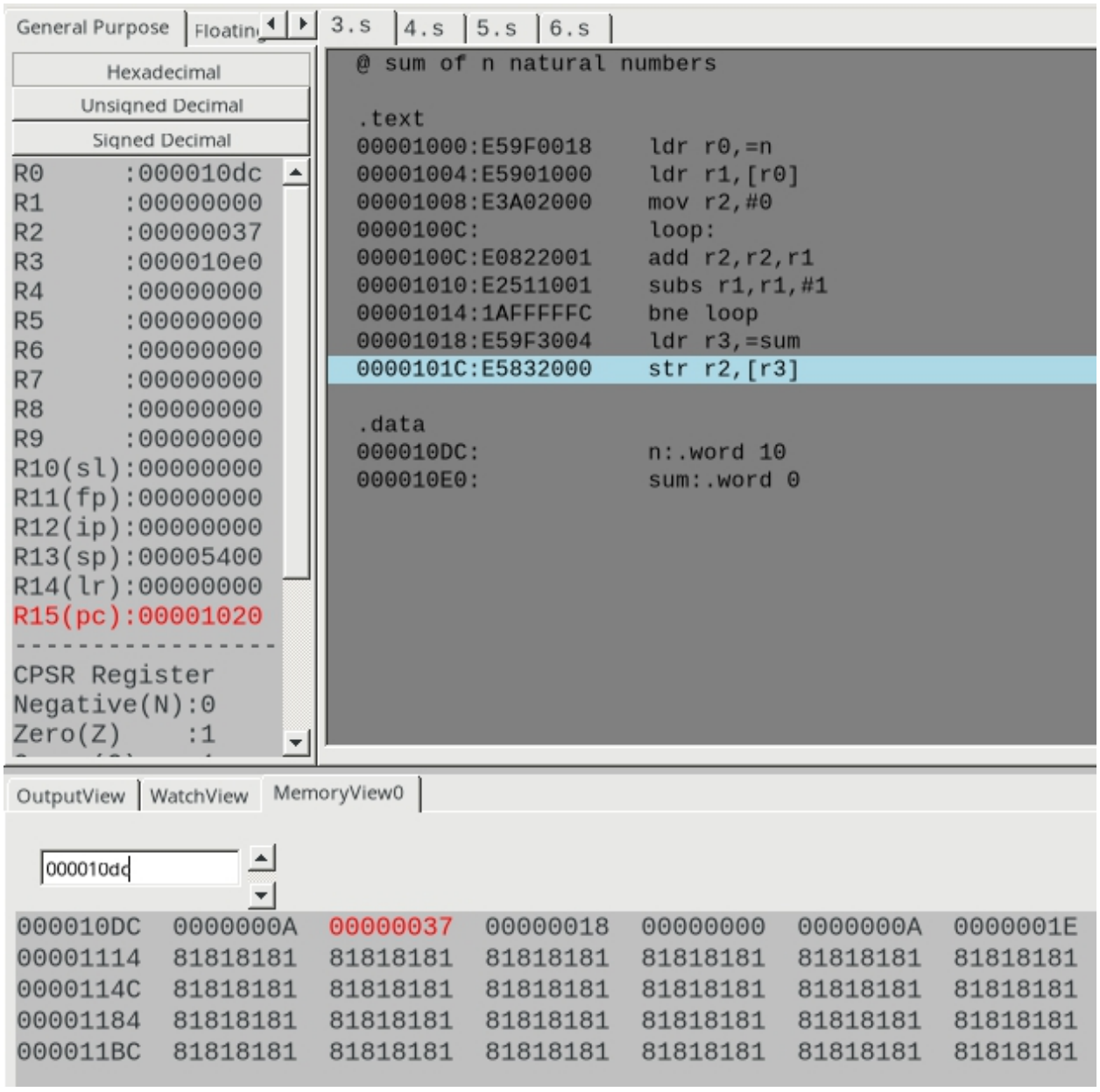
Write a program in ARM7TDMI-ISA to find the sum of N natural numbers. Store the result in the memory location.

I. ARM Assembly Code

Code:

```
@ sum of n natural numbers
.text
        ldr r0,=n
        ldr r1,[r0]
        mov r2,#0
        loop:
                add r2,r2,r1
                subs r1,r1,#1
                bne loop
        ldr r3,=sum
        str r2,[r3]



.data
        n:.word 10
        sum:.word 0
```

II. Output Screen Shots (One)



Week#____2_____ Program Number: ____4___

Title of the Program

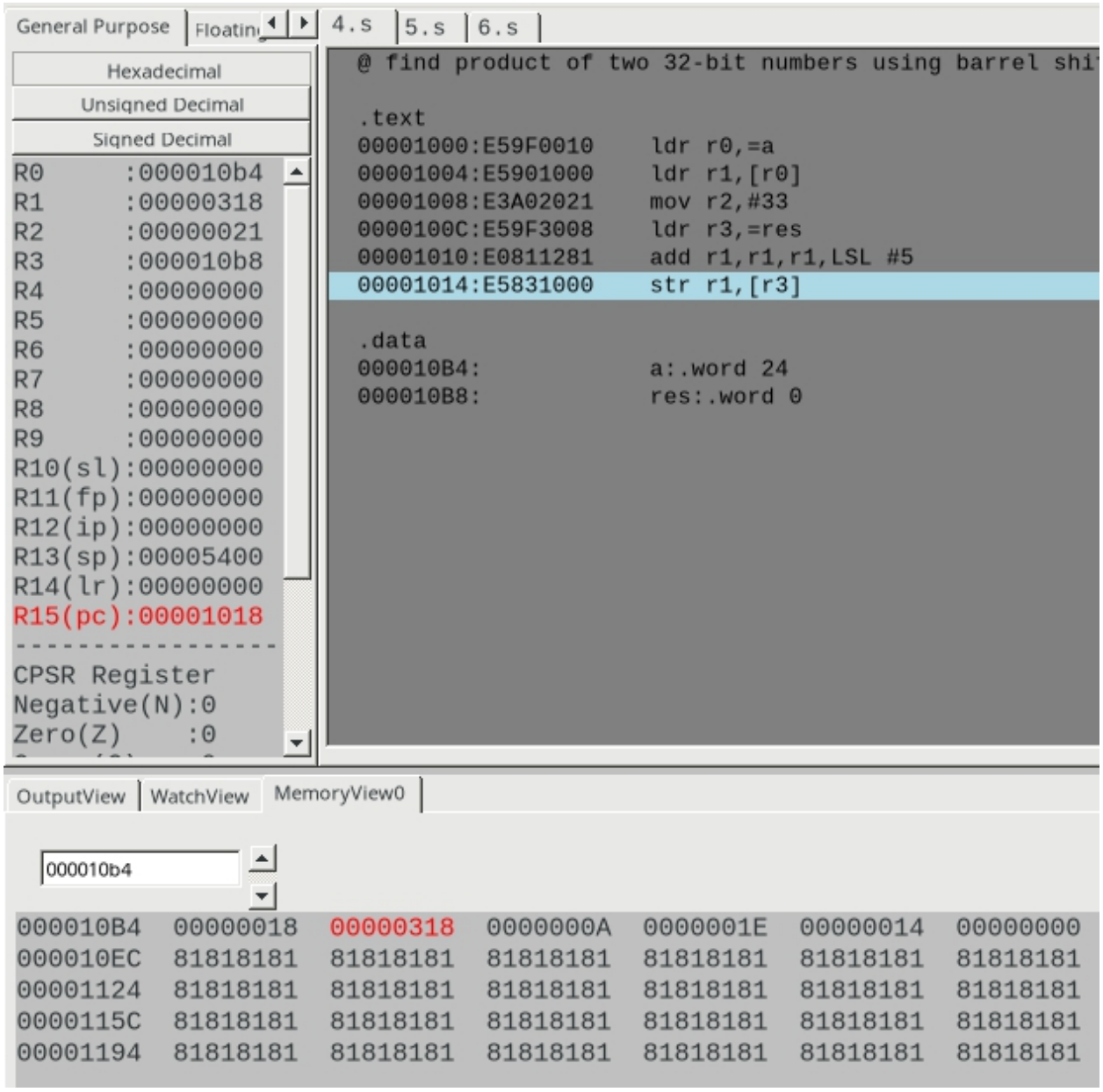
Write a program in ARM7TDMI-ISA to find the product of two 32bit numbers using barrel shifter.

I. ARM Assembly Code

Code:

```
@ find product of two 32-bit numbers using barrel shifter
(here we multiply 24 and 33)


.text
        ldr r0,=a
        ldr r1,[r0]
        mov r2,#33
        ldr r3,=res
        add r1,r1,r1,LSL #5
        str r1,[r3]


.data
        a:.word 24
        res:.word 0
```

II.Output Screen Shot (One)



Week#____2_____ Program Number: ____5___

Title of the Program

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.

I. ARM Assembly Code

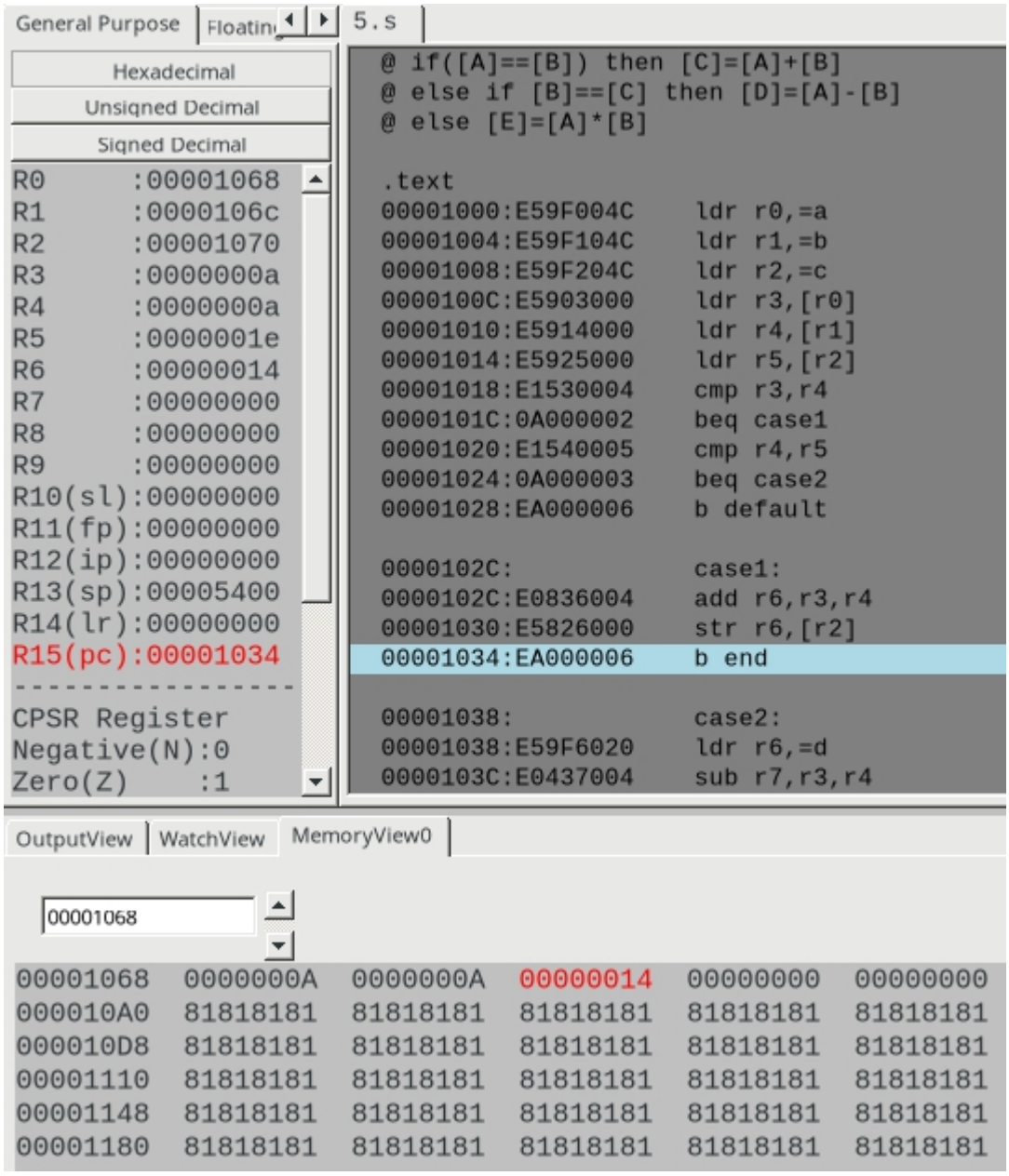
Code:

```
@ if([A]==[B]) then [C]=[A]+[B]
@ else if [B]==[C] then [D]=[A]-[B]
@ else [E]=[A]*[B]
.text
        ldr r0,=a
        ldr r1,=b
        ldr r2,=c
        ldr r3,[r0]
        ldr r4,[r1]
        ldr r5,[r2]
        cmp r3,r4
```

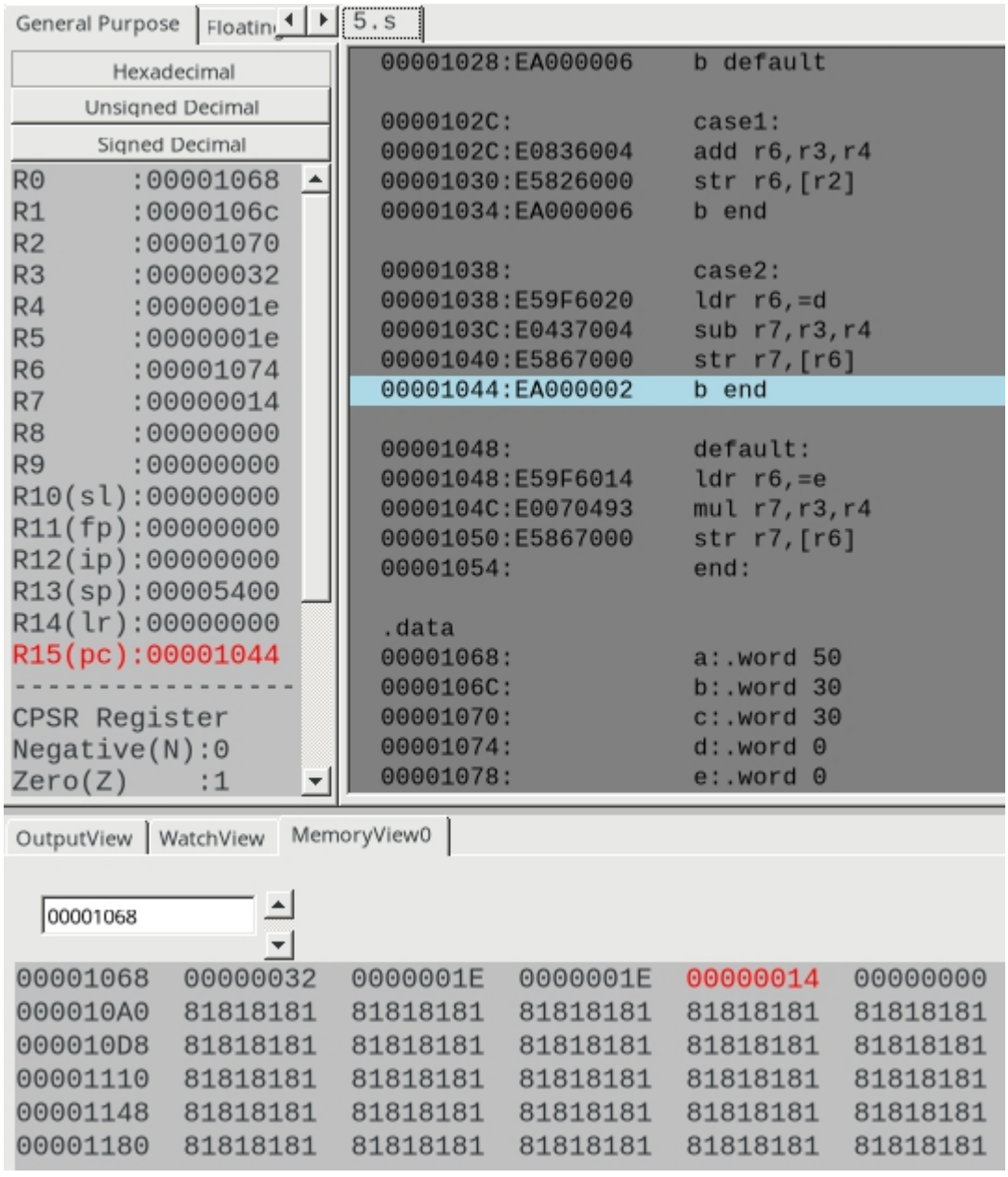
```
        beq case1
        cmp r4,r5
        beq case2
        b default
        case1:
                add r6,r3,r4
                str r6,[r2]
                b end
        case2:
                ldr r6,=d
                sub r7,r3,r4
                str r7,[r6]
                b end
        default:
                ldr r6,=e
                mul r7,r3,r4
                str r7,[r6]
        end:

.data
        a:.word 10
        b:.word 30
        c:.word 20
        d:.word 0
        e:.word 0
```

II.Output Screen Shot (One)

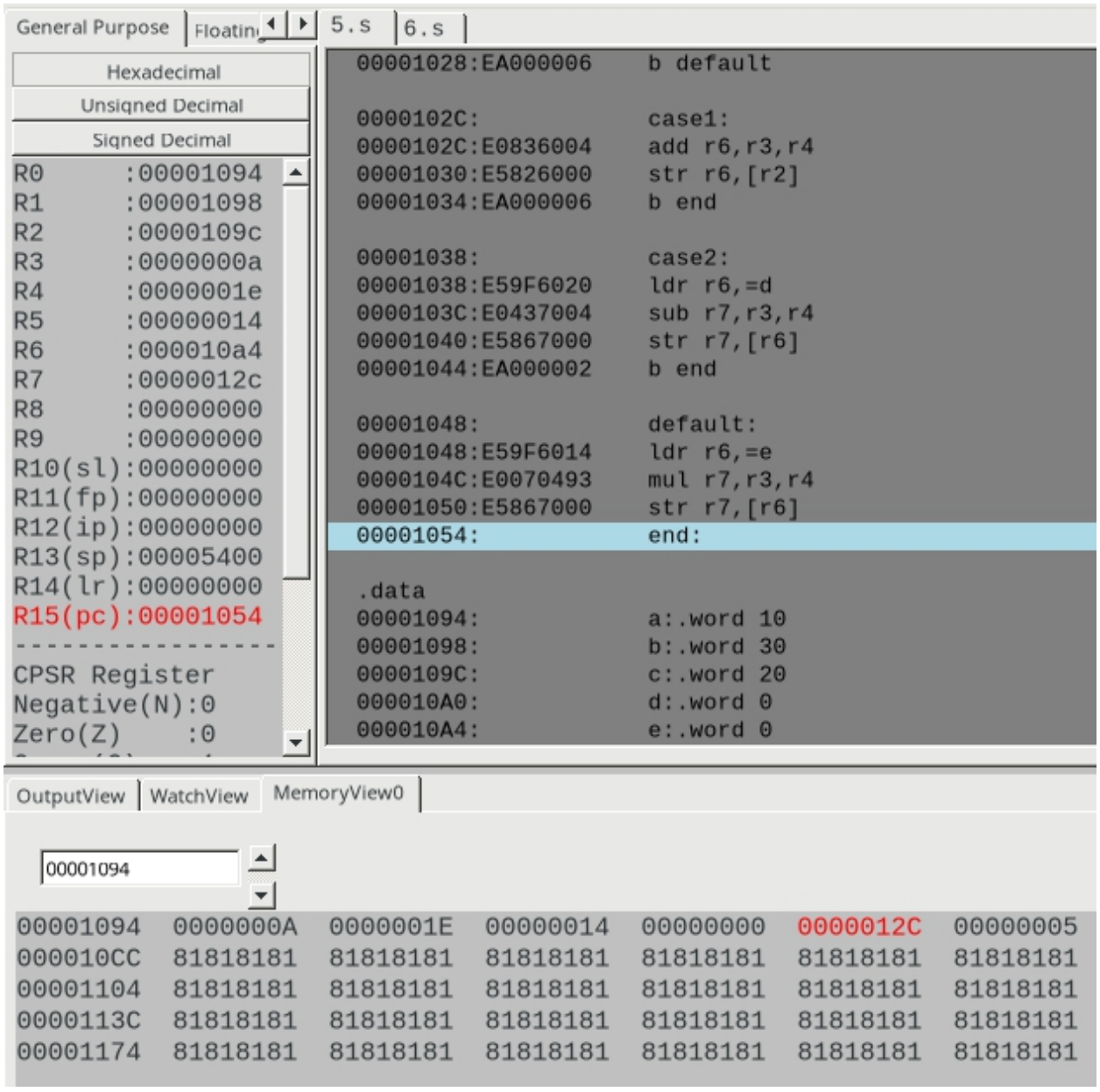
case1: when a==b



case2: when b==c



default:


General Purpose
Floatin
5.s
6.s
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00001094
R1 :00001098
R2 :0000109c
R3 :0000000a
R4 :0000001e
R5 :00000014
R6 :000010a4
R7 :0000012c
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001054
CPSR Register
Negative(N):0
Zero(Z) :0
00001028:EA000006 b default
0000102C: case1:
0000102C:E0836004 add r6,r3,r4
00001030:E5826000 str r6,[r2]
00001034:EA000006 b end
00001038: case2:
00001038:E59F6020 ldr r6,=d
0000103C:E0437004 sub r7,r3,r4
00001040:E5867000 str r7,[r6]
00001044:EA000002 b end
00001048: default:
00001048:E59F6014 ldr r6,=e
0000104C:E0070493 mul r7,r3,r4
00001050:E5867000 str r7,[r6]
00001054: end:
.data
00001094: a:.word 10
00001098: b:.word 30
0000109C: c:.word 20
000010A0: d:.word 0
000010A4: e:.word 0
OutputView
WatchView
MemoryView0
00001094
00001094 0000000A 0000001E 00000014 00000000 0000012C 00000005
000010CC 81818181 81818181 81818181 81818181 81818181 81818181
00001104 81818181 81818181 81818181 81818181 81818181 81818181
0000113C 81818181 81818181 81818181 81818181 81818181 81818181
00001174 81818181 81818181 81818181 81818181 81818181 81818181

Week#____2_____ Program Number: ____6__

Title of the Program

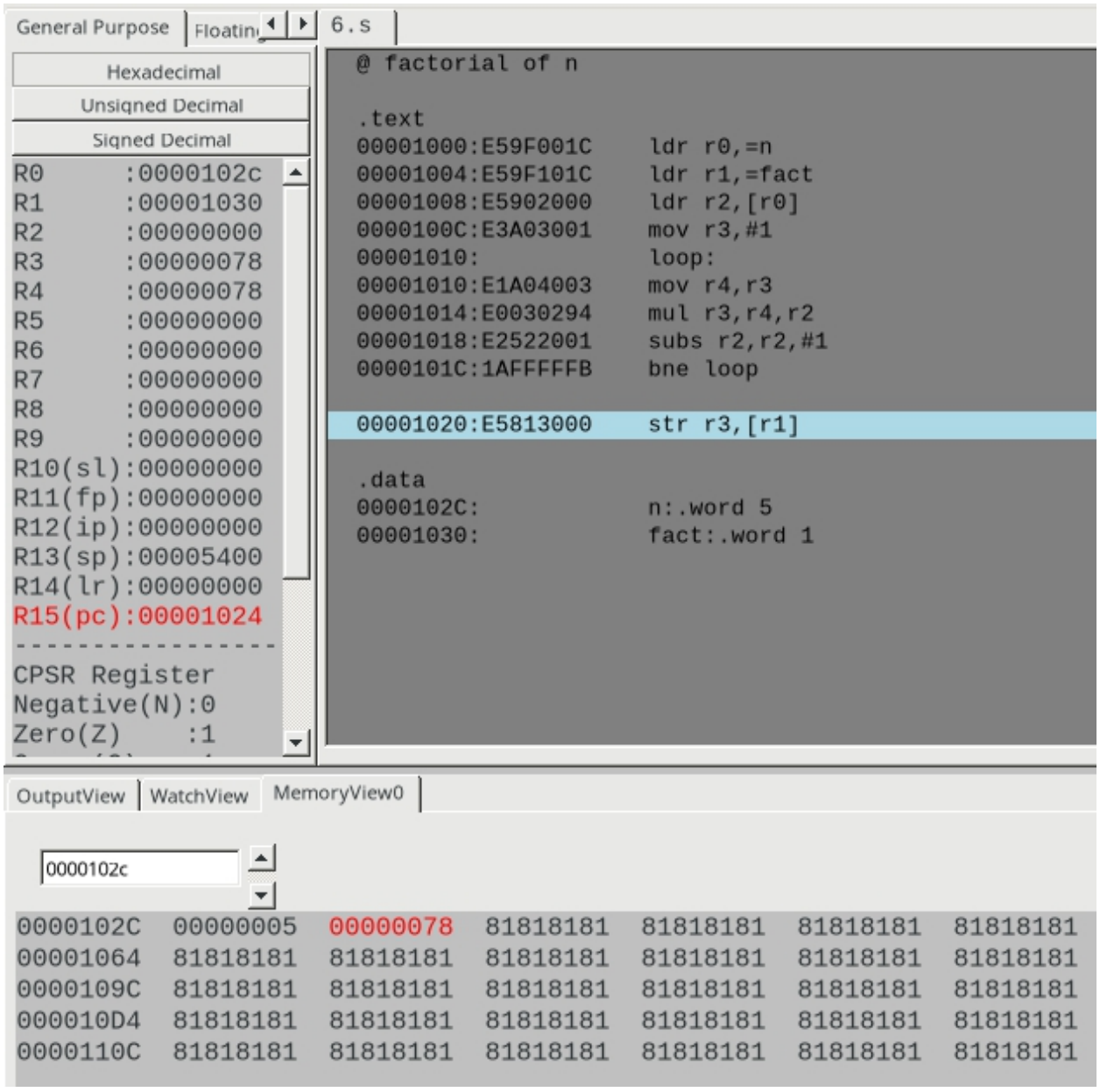
Write a program in ARM7TDMI-ISA to find the factorial of a number.

I. ARM Assembly Code

Code:

```
@ factorial of n
.text
	ldr r0,=n
	ldr r1,=fact
	ldr r2,[r0]
	mov r3,#1
	loop:
		mov r4,r3
		mul r3,r4,r2
		subs r2,r2,#1
		bne loop

	str r3,[r1]

.data
	n:.word 5
	fact:.word 1
```

II.Output Screen Shot (One)



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:
Name: Nihal T M
SRN: PES2UG21CS333
Section: F
Date: 25/01/2023