Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:06/03/2024

Name: Ankith Gowda	SRN: PES2UG22CS077	Section: B

LAB #____5_____   Program Number: ____1___

Title of the Program

1. Write an ALP to multiply 2 matrices.

ARM Assembly Code :

.data

        a: .word 1,2,3,4,5,6,7,8,9   b:

        .word 1,1,2,2,3,3,4,4,5   c:

        .word 0,0,0,0,0,0,0,0,0

.text

```
    LDR R1,=a

    LDR R2,=b

    LDR R3,=c      MOV

R7,#3

MOV R9,#3


    LOOP_3:

        MOV R9,#3

        LDR R2, =b

        B LOOP_2




    LOOP_2:

        CMP R9,#0

        BEQ LOOP_3

        MOV R6,R1

        MOV R5,R2

        MOV R4,#3

        MOV R8,#0

        B LOOP_1


        LOOP_1:
```

```
        LDR R10,[R6]

        ADD R6,R6,#4

        LDR R11,[R5]

        ADD R5,R5,#12        MLA R8,R10,R11,R8        SUB R4,R4,#1        CMP

R4,#0

        BNE LOOP_1

        ADD R2,R2,#4

        STR R8,[R3],#4

SUB R9,R9,#1        CMP

R9,#0


        BNE LOOP_2

        ADD R1,R1,#12

        SUB R7,R7,#1

CMP R7,#0

        BNE LOOP_2

        B END_OP

            END_OP:

        SWI 0X11

.end
```
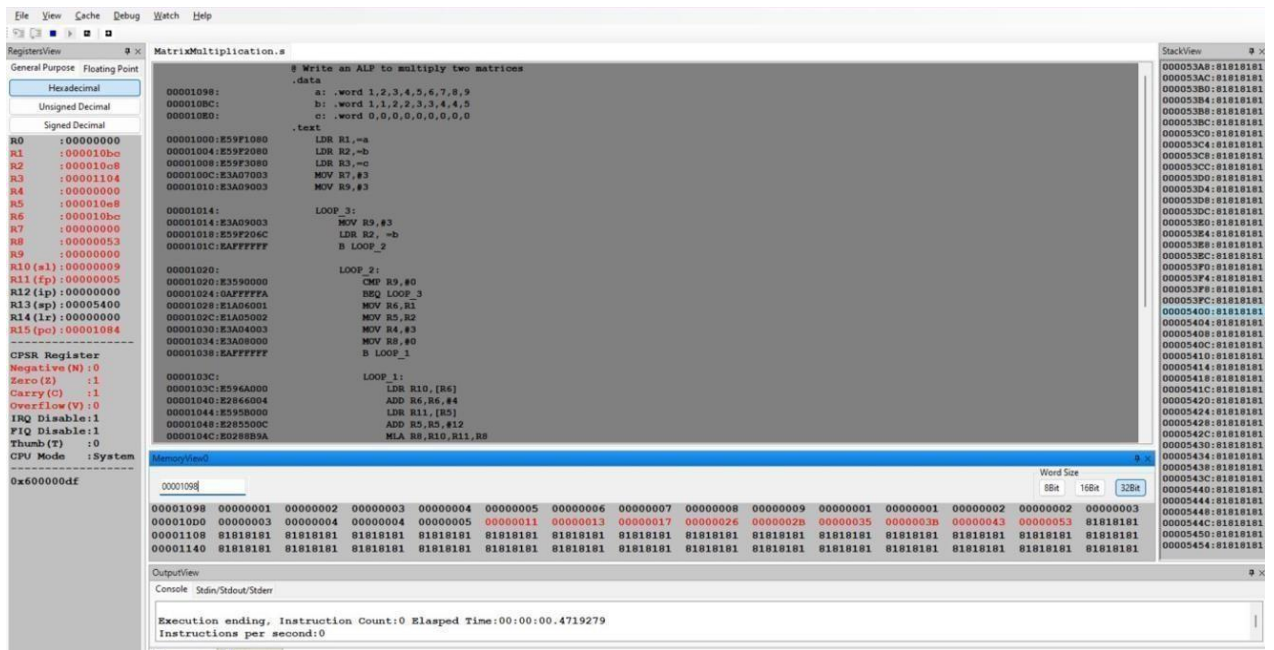
Output Screen Shots *(One Screenshots including Register Window,Memory Window
 and Code Window)*

5_____   Program Number: ____2___

Write an ALP using conditional ARM instructions to sort an array of numbers using Bubble Sort Algorithm.

ARM Assembly Code
CODE:

@Write an ALP to code for Bubble sort

.data

A: .word 12, 10, 3, 8, 7, 4, 1

.text

```
MOV R3, #6 LOOP1:

    LDR R1,=A      MOV

R2, R1

MOV R7, R3


LOOP2:

    LDR R4,[R1]

    ADD R6, R1, #4

    LDR R5, [R6]

    CMP R4, R5

    BMI SKIP

STR R5, [R1]

STR R4, [R6]


    SKIP:

    SUBS R7, R7, #1            ADD R1, R1, #4

    BNE LOOP2

    SUBS R3, R3, #1

    BNE LOOP1

SWI 0X11

.end
```
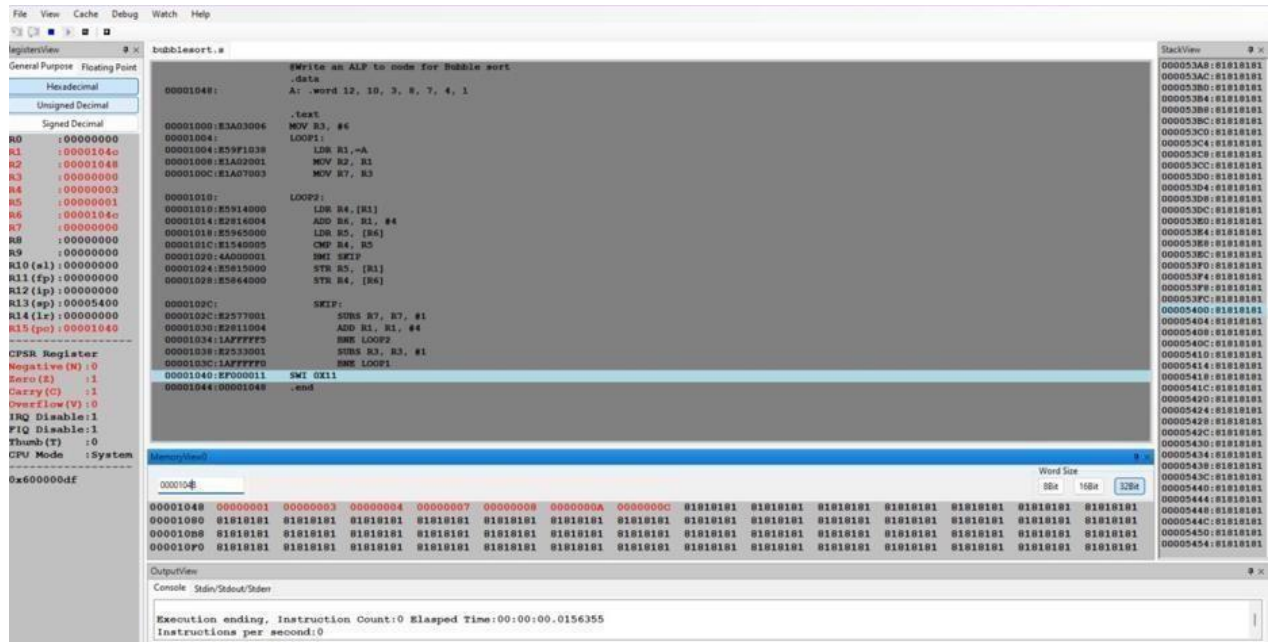
Output Screen Shot *(One Screenshot including Register Window, Memory Window and Code Window)*

5_____   Assignment Question: 1 Write an Assembly Language program to swap the first and last character of a given string.

LAB #____

Title of the Program   ARM

Assembly Code   CODE:

.data   A: .asciz

"Ankith\n"

.text

  LDR R0, =A   SWI

0X02

    MOV R1, #5

    LDRB R2, [R0]

    LDRB R3, [R0, R1]
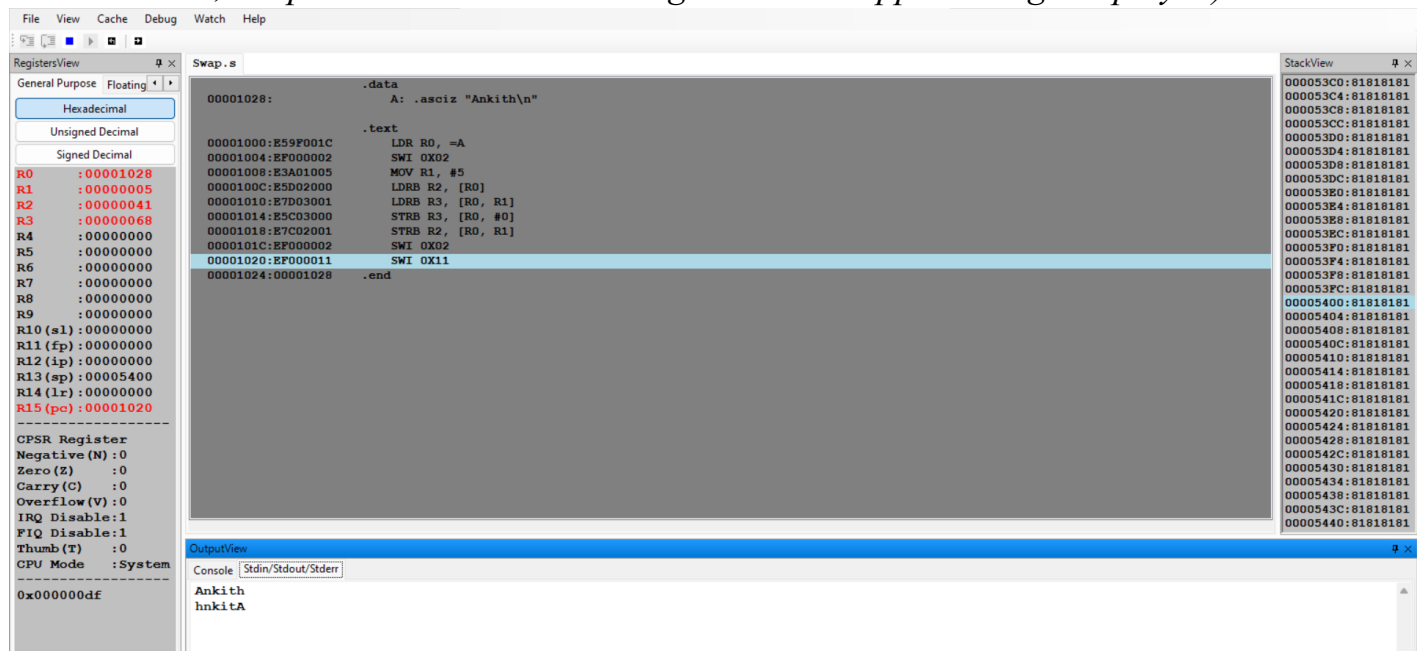
    STRB R3, [R0, #0]

    STRB R2, [R0, R1]

    SWI 0X02

SWI 0X11

.end

Output Screen Shot*(One Screenshot including Register Window,Memory Window and Code Window, Output Window with both original and swapped strings displayed)*

5_____ Assignment Question:____2____ Given a c Code convert it in its equivalent Arm Code. a)x =

(a + b) - c; Program:

LAB #____
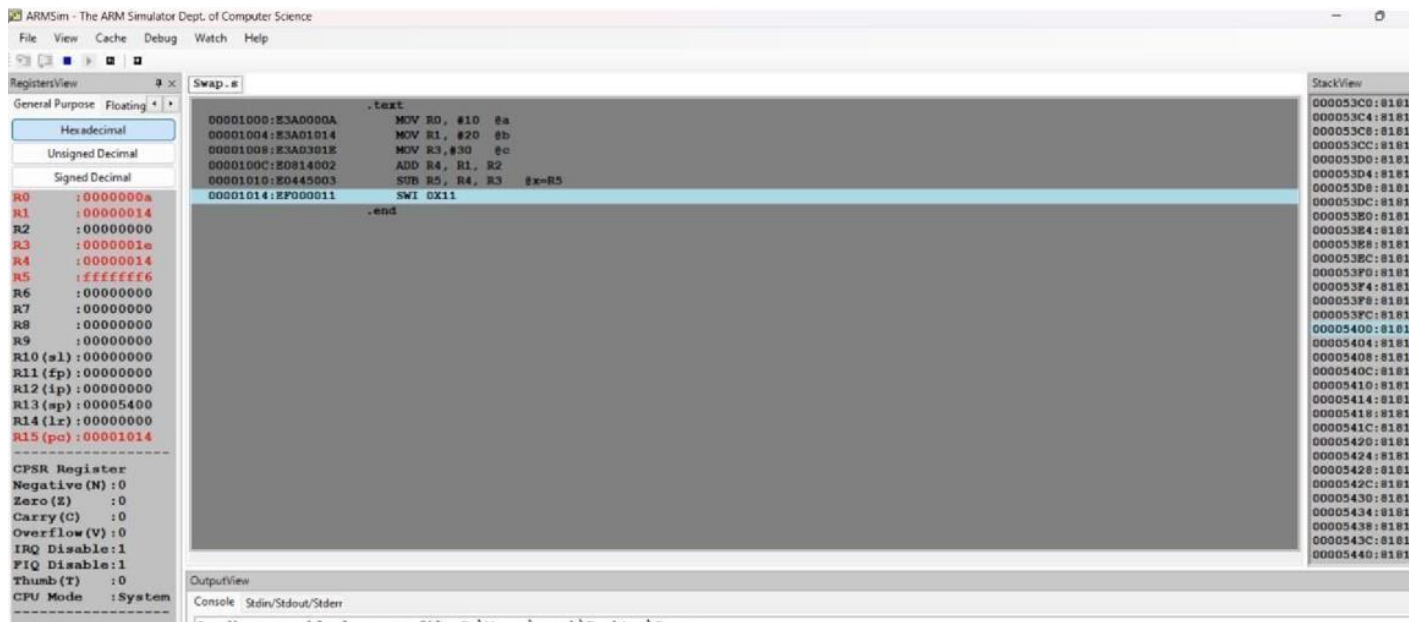
Title of the Program CODE:

.text

  MOV R0, #10 @a

  MOV R1, #20 @b

  MOV R3,#30 @c

  ADD R4, R1, R2

  SUB R5, R4, R3 @x=R5

  SWI 0X11 .end Screenshot:

b)

z = (a << 2) |(b & 15); Program:

CODE:

.text

   MOV R1,#5   @a

   MOV R2,#10 @b

   MOV R3,#0   @z

   MOV R4,R1

   MOV R5,R2

   MOV R4,R4,LSL #2

   AND R5,R5,#15

   ORR R3,R4,R5 @res

   SWI 0X11

.end

Screenshot:

ARMSim - The ARM Simulator Dept. of Computer Science

File   View   Cache   Debug   Watch   Help

RegistersView

General Purpose   Floating

| Hexadecimal |
| Unsigned Decimal |
| Signed Decimal |

R0   :00000000
R1   :00000005
R2   :0000000a
R3   :0000001e
R4   :00000014
R5   :0000000a
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):00001020

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   :System

Swap.s

```
                        .text
00001000:E3A01005       MOV R1,#5     #a
00001004:E3A0200A       MOV R2,#10    #b
00001008:E3A03000       MOV R3,#0     #z
0000100C:E1A04001       MOV R4,R1
00001010:E1A05002       MOV R5,R2
00001014:E1A04104       MOV R4,R4,LSL #2
00001018:E205500F       AND R5,R5,#15
0000101C:E1843005       ORR R3,R4,R5  #res
00001020:EF000011       SWI 0X11
                        .end
```

OutputView

StackView

000053C0:81
000053C4:81
000053C8:81
000053CC:01
000053D0:81
000053D4:81
000053D8:81
000053DC:81
000053E0:81
000053E4:81
000053E8:81
000053EC:81
000053F0:81
000053F4:81
000053F8:81
000053FC:81
00005400:81
00005404:81
00005408:81
0000540C:81
00005410:81
00005414:81
00005418:81
0000541C:81
00005420:81
00005424:81
00005428:81
0000542C:01
00005430:81
00005434:81
00005438:81
0000543C:81
00005440:81