Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E
Course: Computer Organization

Term: Jan to May 2019
Course Code: CS45

Activity IV: Executing ARM programs using ARMsim simulator.

Name: MANAS.P.S	**Marks: /10**	**Date:** 11/2/2020
USN: 1MS18CS065	**Signature of the Faculty:**	

Objective: To simulate ARM Instruction set using ARMsim simulator.

Simulator Used: ARMSim 1.91 is a desktop application running in a Windows environment. It allows users to simulate the execution of ARM assembly language programs on a system based on the ARM7TDMI processor.

ARM enables the users both to debug ARM assembly programs and to monitor the state of the system while a program executes.

Activity to be performed by students:

1) Write an ARM program to generate Fibonacci Series.
2) Write an ARM to search an element in an array and print Y if found and print N if not found.
3) Write an ARM program to find the length of a string and copying one string to another.

Attached in datasheet

Results/Conclusions and Snapshots: Take the snap shot of registers file and memory view

Attached in datasheet

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	MANAS. P.S	Branch:	CSE
USN/Roll No. :	1MS18CS065	Sem/Sec:	IV 'B'
Subject :	Computer Organization and Architecture	Subject Code:	CS45

(1) Write an ARM program to generate Fibonacci series

```
mov  r0,#0
mov  r1,#1
mov  r2,#20
mov  r3,#0
ldr  r4,=0x00002000
mov  r5,#0
loop :  str r0,[r4,r5]
        add r6,r0,r1
        mov  r0,r1
        mov  r1,r6
        add  r5,r5,#4
        add  r3,r3,#1
        cmp r3,r2
        blt loop
swi 0x22
swi 0x11
```

(2) Write an ARM program to search an element in an array and print y if found and print n if not found

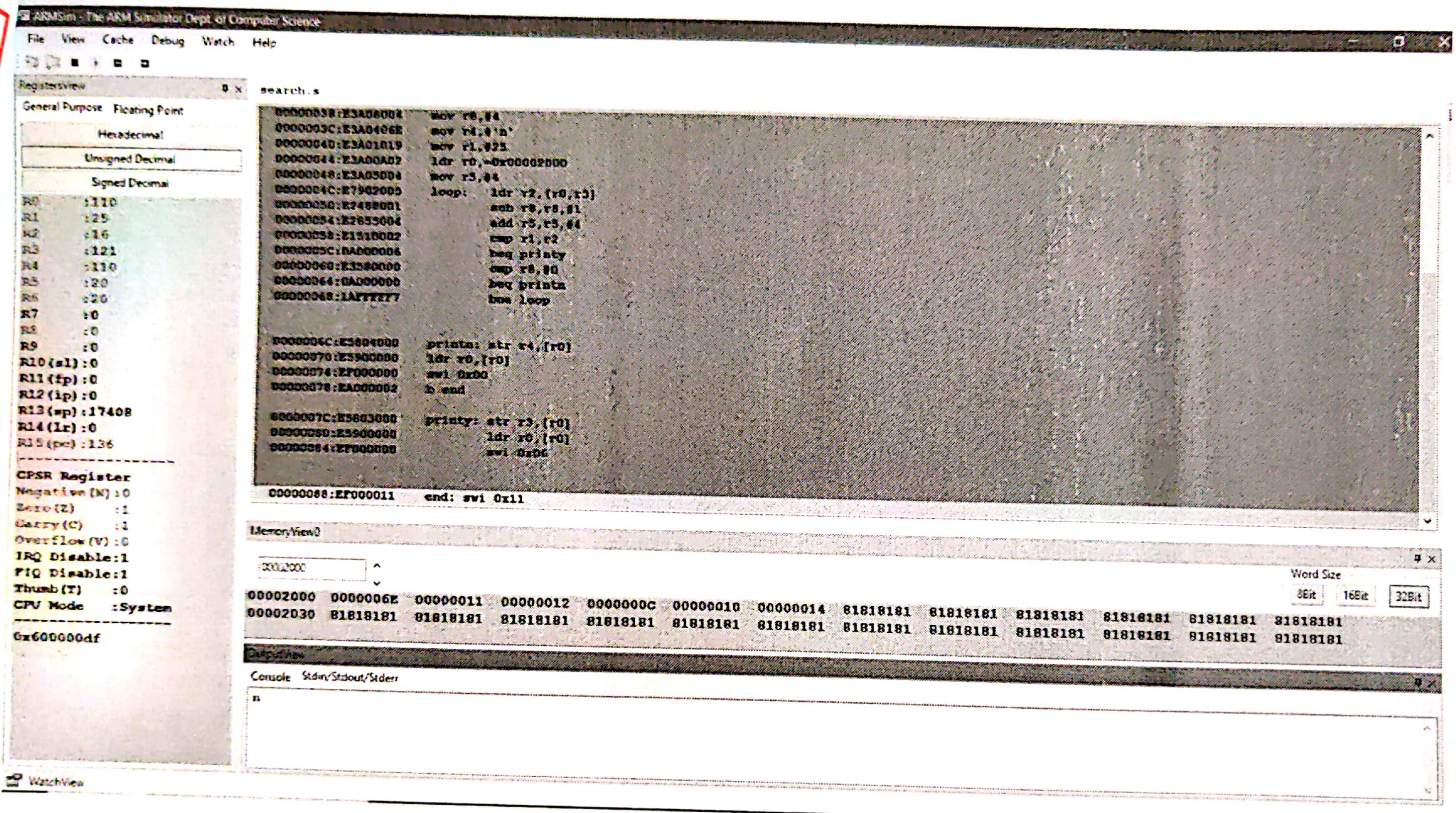
```
ldr r0,=0x00002000
mov r1,#14
mov r2,#17
mov r3,#18
mov r4,#12
mov r5,#16
mov r6,#20
str r1,[r0]
str r2,[r0,#4]
str r3,[r0,#8]
str r4,[r0,#12]
str r5,[r0,#16]
str r6,[r0,#20]
mov r3,#'y'
mov r8,#4
mov r4,#'n'
mov r1,#16
ldr r0,=0x00002000
mov r5,#4
loop: ldr r2,[r0,r5]
      sub r8,r8,#1
      add r5,r5,#4
      cmp r1,r2
      beq printy
      cmp r8,#0
      beq printn
      bne loop

printn : str r4,[r0]
         ldr r0,[r0]
         swi 0x00
         b end
printy : str r3,[r0]
         ldr r0,[r0]
         swi 0x00

end : swi 0x11
```

③ Write an ARM program to find the length of a string and copying one string to another

```
.equ SWI_Open ,0x66
.equ SWI-Close, 0X68
.equ SWI_PrInt, 0x6b
.equ SWI_RdInt, 0x6c
.equ Stdout, 1
.equ SWI_PrStr, 0X69
.equ SWI-Exit,0x11
.global -start
.text

_start :
ldr r0, =FileName
mov r1,#0
swi SWI_Open
bcs Exit
mov r9,r0
mov r5,#0

loopstart :
mov r7,#Stdout
mov r0,r9
ldr r8,=Array
swi SWI_RdInt
bcs afterloop
str r0,[r8,r5]
```

```
add r5,r5,#4
mov r1,r0
mov r0,#Stdout
mov r1,r0
swi SWI_PrInt
add r4,r4,#1
mov r0,#Stdout
ldr r1,=NewLine
swi SWI_PrStr
bal loopstart

afterloop:
mov r5,#20
loop: ldr r2,[r8,r5]
      sub r4,r4,#1
      sub r5,r5,#4
      mov r1,r2
      mov r0,#Stdout
      swi SWI_PrInt
      ldr r1,=NewLine
      swi SWI_PrStr
      cmp r4,#0
      beq end
      bne loop
end : mov r0,r9
      swi SWI_close
```

```
Exit :
swi SWI_Exit

.data
Array :
.align
FileName : .asciz "input.txt"
InFileError : .asciz "Unable to open input file \n"
Newline : .asciz "\n"
.end
```



ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
search.s
MemoryView0
Word Size
8Bit 16Bit 32Bit
OutputView
Console Stdin/Stdout/Stderr
WatchView

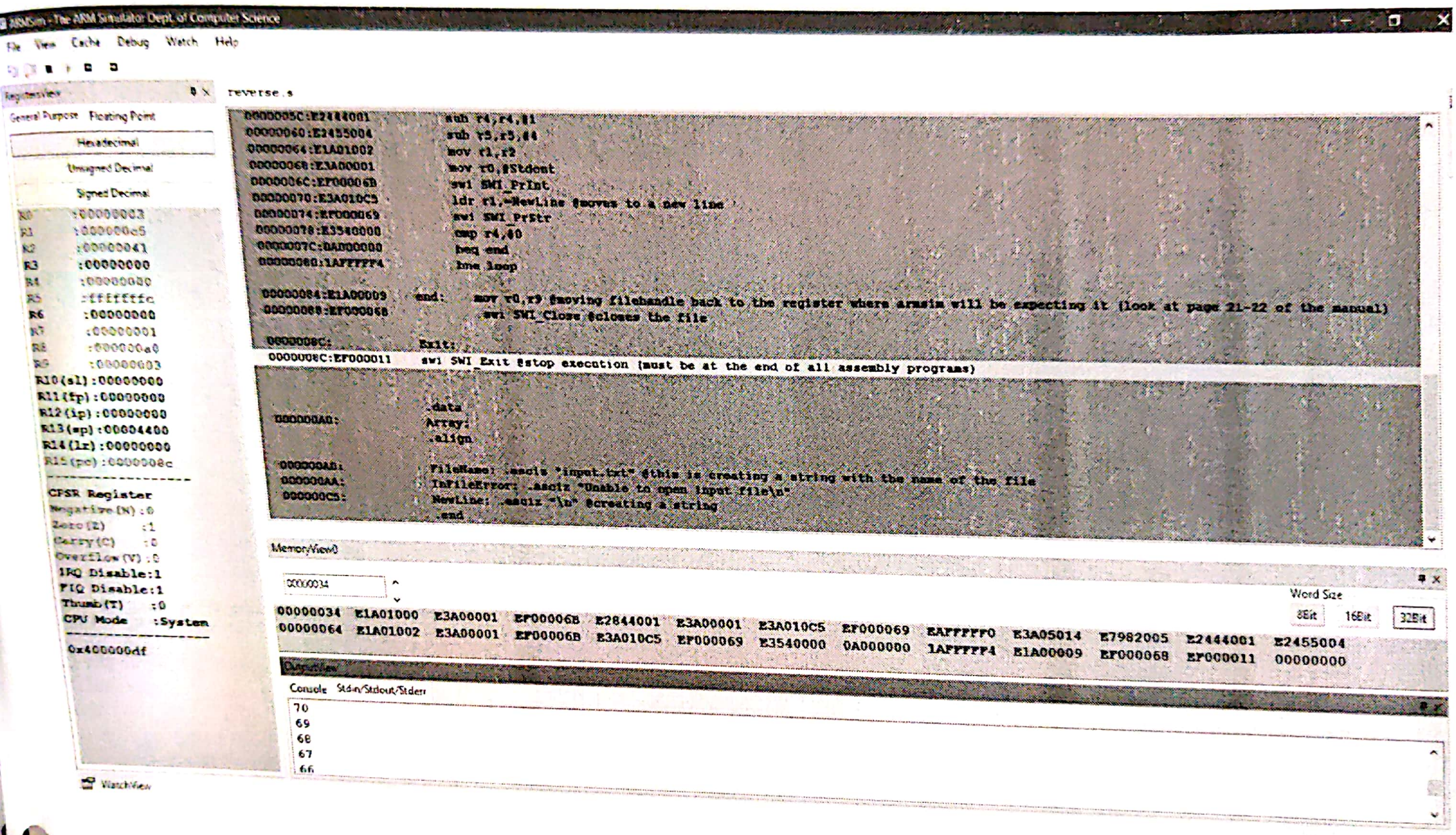
```
R0        :110
R1        :25
R2        :16
R3        :121
R4        :110
R5        :20
R6        :20
R7        :0
R8        :0
R9        :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):17408
R14(lr):0
R15(pc):136
------------------------
CPSR Register
Negative(N):0
Zero(Z)      :1
Carry(C)     :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)     :0
CPU Mode     :System
------------------------
0x600000df
```

```
00000038:E3A08004    mov r8,#4
0000003C:E3A0406E    mov r4,#'n'
00000040:E3A01019    mov r1,#25
00000044:E3A00A02    ldr r0,=0x00002000
00000048:E3A05004    mov r5,#4
0000004C:E7902005    loop:    ldr r2,[r0,r5]
00000050:E2488001             sub r8,r8,#1
00000054:E2855004             add r5,r5,#4
00000058:E1510002             cmp r1,r2
0000005C:0A000006             beq printy
00000060:E3580000             cmp r8,#0
00000064:0A000000             beq printn
00000068:1AFFFFF7             bne loop


0000006C:E5804000    printn: str r4,[r0]
00000070:E5900000    ldr r0,[r0]
00000074:EF000000    swi 0x00
00000078:EA000002    b end

0000007C:E5803000    printy: str r3,[r0]
00000080:E5900000            ldr r0,[r0]
00000084:EF000000            swi 0x00

00000088:EF000011    end: swi 0x11
```

```
00002000
00002000  0000006E  00000011  00000012  0000000C  00000010  00000014  81818181  81818181  81818181  81818181  81818181  81818181
00002030  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
n
```

reverse-proper.PNG


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
reverse.s
MemoryView0
Word Size
8Bit 16Bit 32Bit
OutputView
Console Stdin/Stdout/Stderr
WatchView

```
R0       :00000003
R1       :000000c5
R2       :00000041
R3       :00000000
R4       :00000000
R5       :fffffffc
R6       :00000000
R7       :00000001
R8       :000000a0
R9       :00000003
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00004400
R14(lr) :00000000
R15(pc) :0000008c
--------------------
CPSR Register
Negative(N) :0
Zero(Z)     :1
Carry(C)    :0
Overflow(V) :0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
--------------------
0x400000df
```

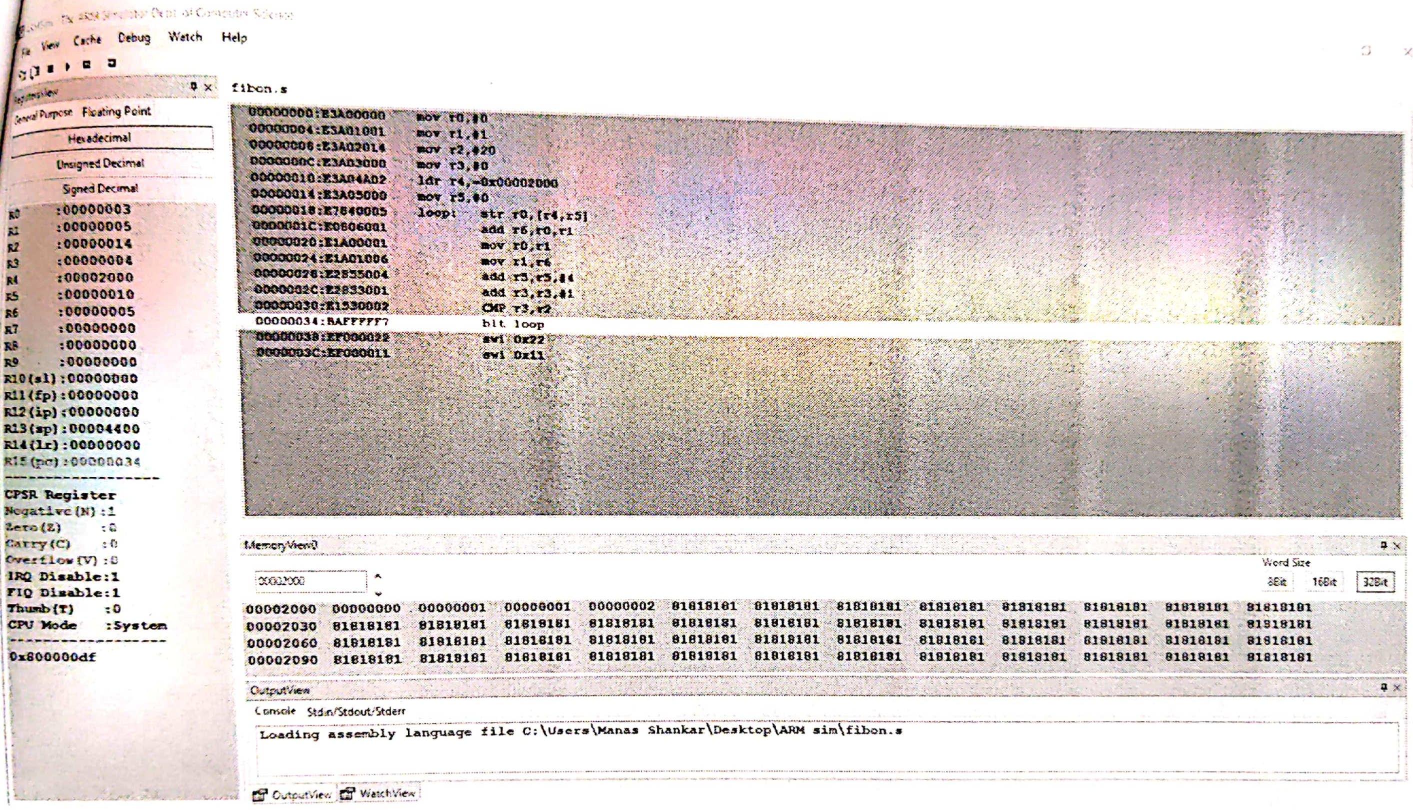
```
0000005C:E2444001          sub r4,r4,#1
00000060:E2455004          sub r5,r5,#4
00000064:E1A01002          mov r1,r2
00000068:E3A00001          mov r0,#Stdout
0000006C:EF00006B          swi SWI_PrInt
00000070:E3A010C5          ldr r1,=NewLine @moves to a new line
00000074:EF000069          swi SWI_PrStr
00000078:E3540000          cmp r4,#0
0000007C:0A000000          beq end
00000080:1AFFFFF4          bne loop

00000084:E1A00009   end:    mov r0,r9 @moving filehandle back to the register where armsim will be expecting it (look at page 21-22 of the manual)
00000088:EF000068            swi SWI_Close @closes the file

0000008C:           Exit:
0000008C:EF000011   swi SWI_Exit @stop execution (must be at the end of all assembly programs)

                    .data
000000A0:           Array:
                    .align

000000A0:           FileName: .ascii "input.txt" @this is creating a string with the name of the file
000000AA:           InFileError: .asciz "Unable to open input file\n"
000000C5:           NewLine: .asciz "\n" @creating a string
                    .end
```

```
00000034
00000034  E1A01000  E3A00001  EF00006B  E2844001  E3A00001  E3A010C5  EF000069  EAFFFFF0  E3A05014  E7982005  E2444001  E2455004
00000064  E1A01002  E3A00001  EF00006B  E3A010C5  EF000069  E3540000  0A000000  1AFFFFF4  E1A00009  EF000068  EF000011  00000000
```

```
70
69
68
67
66
```

2020

fibon-1.PNG


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
MemoryView0
Word Size
8Bit 16Bit 32Bit
OutputView
Console Stdin/Stdout/Stderr
OutputView WatchView

```
R0        :00000003
R1        :00000005
R2        :00000014
R3        :00000004
R4        :00002000
R5        :00000010
R6        :00000005
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00004400
R14(lr)   :00000000
R15(pc)   :00000034
--------------------
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    :System
--------------------
0x800000df
```

fibon.s

```
00000000:E3A00000    mov r0,#0
00000004:E3A01001    mov r1,#1
00000008:E3A02014    mov r2,#20
0000000C:E3A03000    mov r3,#0
00000010:E3A04A02    ldr r4,=0x00002000
00000014:E3A05000    mov r5,#0
00000018:E7840005    loop:   str r0,[r4,r5]
0000001C:E0806001            add r6,r0,r1
00000020:E1A00001            mov r0,r1
00000024:E1A01006            mov r1,r6
00000028:E2855004            add r5,r5,#4
0000002C:E2833001            add r3,r3,#1
00000030:E1530002            CMP r3,r2
00000034:BAFFFFF7            blt loop
00000038:EF000022            swi 0x22
0000003C:EF000011            swi 0x11
```

00002000

```
00002000  00000000  00000001  00000001  00000002  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00002030  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00002060  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00002090  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Loading assembly language file C:\Users\Manas Shankar\Desktop\ARM sim\fibon.s
```