

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

Tutorial -III

Programme: B.E
Course: Computer Organization Course Code: CS45

Term: Jan to May 2020

Name: <u>MANAB. P.S</u>	Marks: <u>19/10</u>	Date: <u>4/2/2020</u>
USN: <u>1MS18CS065</u>	Signature of the Faculty: <u>[Signature]</u>	<u>10/2/2020</u>

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 perform basic arithmetic operations.
- 2) Write an ARM program to demonstrate the working of load and store instructions.
- 3) Write an ARM program to evaluate expression $f=(g+h)-(i+j)$
- 4) Write an ARM program to find the sum of all elements of an array.
- 5) Write an ARM program to find the factorial of a number.

Programs and the snapshots:



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

① Write an ARM program to perform basic arithmetic operations.

```
MOV R3, #24
MOV R4, #25
LDR R1, = 0x00000064
LDR R2, = 0x00000068
```

```
STR R3, [R1]
STR R4, [R2]
```

```
LDR R5, [R1]
LDR R6, [R2]
```

```
ADD R0, R5, R6 ; Basic arithmetic operations
MUL R0, R5, R6
SUB R0, R5, R6
```

```
SWI 0x11
```

② Write an ARM program to perform and demonstrate working of load and store instructions.

```
MOV R3, #36 ; Values to store
MOV R4, #37
```



```

LDR R1, =0x00000074 ; load address into
LDR R2, =0x00000078 ; register

STR R3, [R1] ; store R3 and R4 at address in
STR R4, [R2] ; R1 and R2

LDR R5, [R1] ; load from R1, R2 to R5, R6
LDR R6, [R2]

ADD R0, R5, R6
SUB R0, R5, R6
MUL R0, R5, R6

SWI 0x11

```

③ Write an ARM program to evaluate expression
 $f = (g+h) - (i+j)$

```

MOV R3, #41 ; value of g
MOV R4, #45 ; value of h
MOV R7, #35 ; value of i
MOV R8, #38 ; value of j
LDR R1, =0x00000064
LDR R2, =0x00000068
LDR R9, =0x00000074
LDR R10, =0x00000078

STR R3, [R1]
STR R4, [R2]
STR R7, [R9]
STR R8, [R10]

```

```

LDR R5, [R1]
LDR R6, [R2]
LDR R11, [R9]
LDR R12, [R10]

```

```

ADD R5, R5, R6 ; q+h

```

```

ADD R11, R11, R12 ; i+j

```

~~ADD~~

```

SUB R0, R5, R11 ; (q+h)-(i+j), store result in R0

```

```

SWI 0x11

```

④ Write an ARM program to find sum of all elements in an array

```

MOV R0, #5 ; initialize counter register

```

```

LDR R1, =array ; load base address

```

```

loop: LDR R2, [R1], #4 ; load array val and increment

```

```

      ADD R3, R3, R2 ; address. Sum stored in R3.

```

```

      SUB R0, R0, #1 ; Decrement counter value

```

```

      CMP R0, #0 ; checking counter value

```

```

      BNE loop

```

```

      SWI 0x11

```

• data

```

array: .word 0x00000001, 0x0000000A, 0x000000AB,
          0x0000000B, 0x00000008

```

⑤ Write an ARM program to find the factorial of a number.


```

LDR R1, =fact ; load base address of variables
LDR R2, [R1] ; load value at [R1] into R2 and R3
LDR R3, [R1]
loop: SUB R2, R2, #1 ; decrement R2 value
      CMP R2, #0 ; break to end if R2 == 0
      BEQ end
      MUL R0, R2, R3 ; calculate product in current
      MOV R3, R0 ; iteration, store into R3 for
      CMP R2, #0 ; updation. check R2 value.
      BNE loop ; break if equal to zero
end: SWI 0x11

.data
fact: .word 0x00000005 ; variable with value 5

```

ARMsim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

Registers View

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00000000
R1 : 00000064
R2 : 00000068
R3 : 00000029
R4 : 0000002d
R5 : 00000056
R6 : 0000002d
R7 : 00000023
R8 : 00000026
R9 : 00000074
R10 (sl) : 00000078
R11 (fp) : 00000049
R12 (ip) : 00000026
R13 (sp) : 00004400
R14 (lr) : 00000000
R15 (pc) : 00000048

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

0x000000df

expr.s

```

00000000: E3A03029 MOV R3, #41
00000004: E3A0402D MOV R4, #45
00000008: E3A07023 MOV R7, #35
0000000C: E3A08026 MOV R8, #38
00000010: E3A01064 LDR R1, =0x00000064
00000014: E3A02068 LDR R2, =0x00000068
00000018: E3A09074 LDR R9, =0x00000074
0000001C: E3A0A078 LDR R10, =0x00000078
00000020: E5913006 STR R3, [R1]
00000024: E5924000 STR R4, [R2]
00000028: E5987000 STR R7, [R3]
0000002C: E59AA000 STR R8, [R10]
00000030: E5913006 LDR R3, [R1]
00000034: E5924000 LDR R4, [R2]
00000038: E5988000 LDR R8, [R3]
0000003C: E59AC000 LDR R13, [R10]
00000040: E0835006 ADD R3, R3, R4
00000044: E083800C ADD R11, R11, R12
00000048: E0450008 STR R0, R5, R11
0000004C: E7000011 SWI #0x11

```

Memory View

Word Size 32b 16b 8b

00000024	E5924000	E5987000	E59AA000	E5915000	E5926000	E599B000	E59AC000	E0855006	E08B300C	E0450008	EP000011	01010101	01010101
00000028	01010101	01010101	01010101	01010101	00000029	0000002D	01010101	01010101	00000023	00000026	01010101	01010101	01010101
0000002C	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101
00000030	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101

Output View

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Manas Shankar\Desktop\ARMsim\expr.s

ARMsim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

Registers View

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00000001
R1 : 00000034
R2 : 0000000b
R3 : 000000c1
R4 : 00000000
R5 : 00000000
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (sl) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00004400
R14 (lr) : 00000000
R15 (pc) : 00000018

CPSR Register

Negative (N) : 0
Zero (Z) : 0
Carry (C) : 1
Overflow (V) : 0
IRQ Disable : 1
FIQ Disable : 1
Thumb (T) : 0
CPU Mode : System

0x200000df

array.s

```

00000000: E3A00003 MOV R0, #3
00000004: E3A01024 LDR R1, =array
00000008: E4917004 loop: LDR R2, [R1], #4
0000000C: E0E33002 ADD R3, R3, R2
00000010: E2400001 SUB R0, R0, #1
00000014: E3500000 CMT R0, #0
00000018: 1AFF777A BNE loop
0000001C: EP000011 SWI #0x11

```

.data
array: .word 0x00000001, 0x0000000A, 0x0000000A, 0x00000008, 0x00000008

Memory View

Word Size 32b 16b 8b

00000024	00000001	0000000A	0000000A	00000008	00000008	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101
00000028	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101
0000002C	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101
00000030	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101	01010101

Output View

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Manas Shankar\Desktop\ARMsim\array.s


```
00000000:E3A01020 MOV R1, #0x00000020
00000004:E3A02002 MOV R2, #2
00000008:E3A03001 MOV R3, #1
0000000C:E3A04001 MOV R4, #1
00000010:E0811002 ADD R1,R1,R2
00000014:E0813004 ADD R3,R3,R4
00000018:E0411003 SUB R1,R1,R3
```

```
0000001C:EF000011 SWI 0x11
```

Loading assembly language file C:\Users\IBMORVIDIA-PC-01\Desktop\kaus\EXP.s

```
00000000:E3A01030 MOV R1, #0x00000030
00000004:E3A03000 MOV R3, #0
00000008:E3A04003 MOV R4, #3
0000000C:E7814003 STR R4, [R1,R3]
00000010:E7915003 LDR R5, [R1,R3]
```

```
00000014:EF000011 SWI 0x11
```

Loading assembly language file C:\Users\IBMORVIDIA-PC-01\Desktop\kaus\STORZ.s

