

# Microprocessor and Computer Architecture Laboratory

## UE20CS256

Name: Naman Choudhary	SRN: PES2UG20CS209	Section: D
-----------------------	--------------------	------------

### Week 3

<b>Program Number</b>	1
<b>Program Qn</b>	<p>Write a program in ARM7TDMI-ISA to find the sum of N data items in the memory. Store the result in the memory location.</p> <ul style="list-style-type: none"> <li>a. Use Full word (.word directive)</li> <li>b. Use Half word(.Hword directive)</li> <li>c. Use Byte wise (.Byte directive)</li> </ul> <p>{1,2,3,4,5}</p>

### ARM Assembly Code

```

.DATA
    //Let N=5
    A : .WORD 12,8,5,7,13
    //FULL WORD
    B : .HWORD 12,8,5,7,13
    //HALF WORD
    C : .byte 12,8,5,7,13
    //BYTE WISE

.TEXT
    LDR r1,=A
    LDR r2,=B
    LDR r3,=C
    LDR r9,=C
    ADD r9,r9,#6
    mov r4,#0
        //Stores the count for no of loops
    mov r5,#0
    mov r6,#0
    mov r7,#0

```

L1 :

```

LDR r8, [r1]
//loads the number
ADD r5, r5, r8
//intermediate sum
ADD r1, #4
//next numbers address
ADD r4, #1
//increase counter
CMP r4, #5
//checks if 5 runs
BNE L1
STR r5, [r9]
ADD r9, #4
mov r4, #0
//Resets count to 0

```

L2 :

```

LDRH r8, [r2]
ADD r6, r6, r8
ADD r2, #2
ADD r4, #1
CMP r4, #5
BNE L2
STR r6, [r9]
ADD r9, #4
mov r4, #0
//Resets count to 0

```

L3 :

```

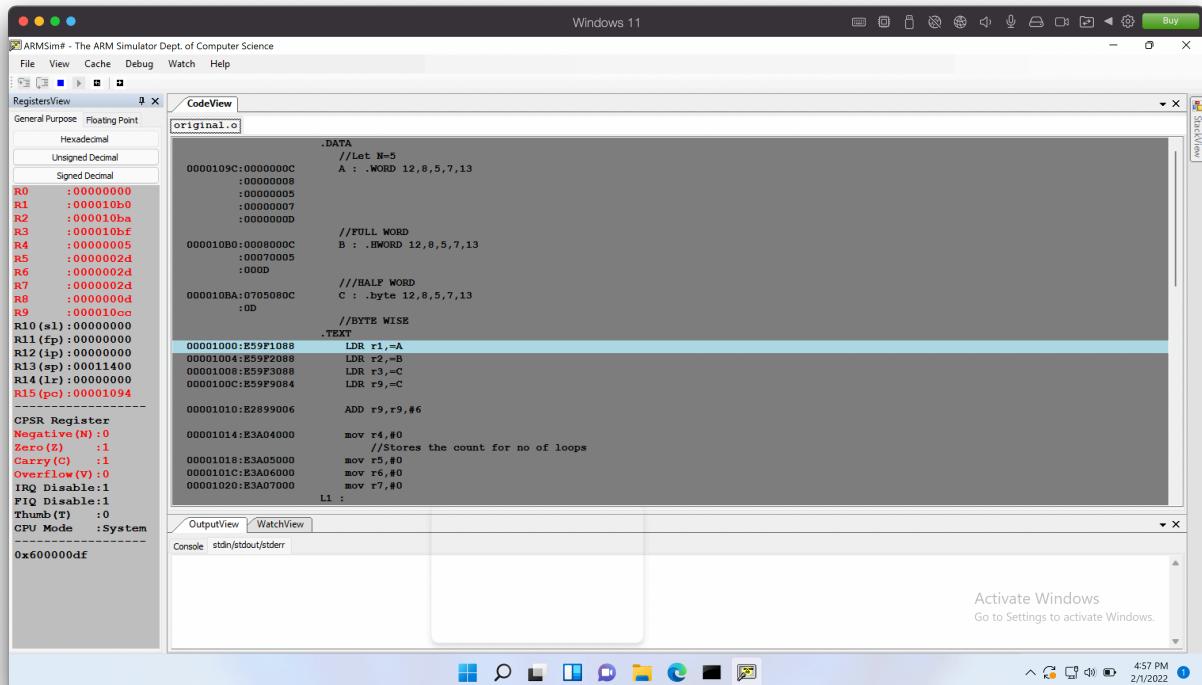
LDRB r8, [r3]
ADD r7, r7, r8
ADD r3, #1

ADD r4, #1
CMP r4, #5
BNE L3
STR r7, [r9]
ADD r9, #4
B L4

```

L4 : .end

## Output Screenshot



<b>Program Number</b>	2
<b>Program Qn</b>	Write a program in ARM7TDMI-ISA to find the sum of N natural numbers. Store the result in the memory location.

## ARM Assembly Code

```
.DATA
```

```
    A : .WORD 11  
    //value of n
```

```
.TEXT
```

```
    LDR r3,=A  
    LDR r0,[r3]  
    //store value  
    mov r1,#0  
    //count  
    mov r2,#0
```

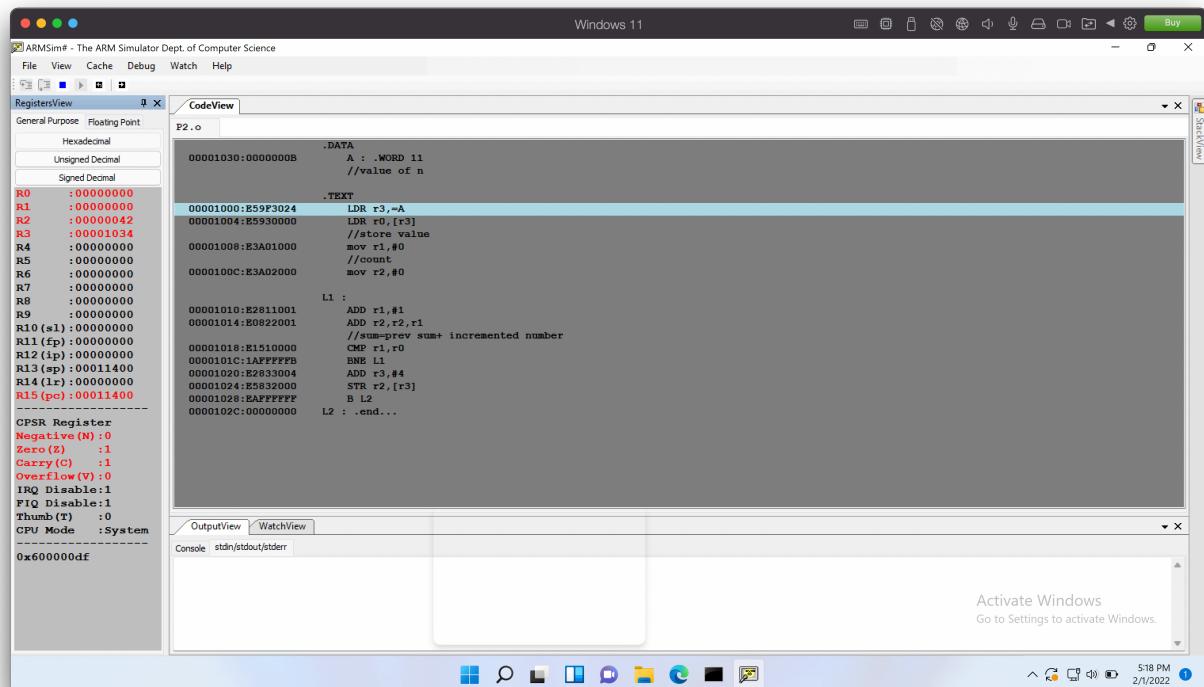
```
L1 :
```

```
    ADD r1,#1  
    ADD r2,r2,r1  
    //sum=prev sum+ incremented number  
    CMP r1,r0  
    BNE L1  
    ADD r3,#4  
    STR r2,[r3]
```

```
    B L2
```

```
L2 : .end
```

## Output Screenshot



<b>Program Number</b>	3
<b>Program Qn</b>	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.
<b>ARM Assembly Code</b>	

```
.DATA
A : .WORD 12
B : .WORD 12
C : .WORD 50
D : .WORD 0
E : .WORD 0
```

```
.TEXT
LDR r0,=A
LDR r1,=B
LDR r2,=C
LDR r3,=D
LDR r4,=E
LDR r5, [r0]
LDR r6, [r1]
LDR r7, [r2]
LDR r9,=C
CMP r5, r6
//If A==B
BEQ L1
CMP r6, r7
//Else If B==C
BEQ L2
B L3
//Else
```

```
L1 :
ADD r8, r5, r6
//C=A+B
STR r8, [r9]
B L4
L2 :
SUB r8, r5, r6
//D=A-B
STR r8, [r3]
B L4
L3 :
MUL r8, r5, r6
//E=A*B
STR r8, [r4]
B L4
//r8 Stores the results
L4 : .end
```

## Output Screenshot

If:

The screenshot shows the ARMSim# interface running on Windows 11. The assembly code in the CodeView window is as follows:

```

P3.o

    .DATA
    00001070:0000000C    A : .WORD 12
    00001074:0000000C    B : .WORD 12
    00001078:00000032    C : .WORD 30
    0000107C:00000000    D : .WORD 0
    00001080:00000000    E : .WORD 0

    .TEXT
    00001000:E59F0054    LDR r0,-A
    00001004:E59F1004    LDR r1,-B
    00001008:E59F2054    LDR r2,-C
    0000100C:E59F3054    LDR r3,-D
    00001010:E59F4054    LDR r4,-E
    00001014:E5900000    LDR r5,[r0]
    00001018:E5916000    LDR r6,[r1]
    0000101C:E5927000    LDR r7,[r2]
    00001020:E59F903C    LDR r9,-C
    00001024:E1550006    CMP r5,r6
                                //If A==B
    00001028:D0000002    BEQ L1
    0000102C:E1560007    CMP r6,r7
                                //Base If B==C
    00001030:0A000003    BEQ L2
    00001034:EAD00005    B L3
                                //Else
    L1 :
    00001038:E0858006    ADD r8,r5,r6
                                //C=A+B

```

The RegistersView window shows the initial state of registers R0-R15. The CPSR Register shows Negative (N) = 0, Zero (Z) = 1, Carry (C) = 1, and Overflow (V) = 0.

Else if:

The screenshot shows the ARMSim# interface running on Windows 11. The assembly code in the CodeView window is identical to the previous one, but the RegistersView window shows different initial values for registers R0-R15. The CPSR Register shows Negative (N) = 0, Zero (Z) = 1, Carry (C) = 1, and Overflow (V) = 0.

**Else:**

The screenshot shows the ARMSim# - The ARM Simulator application running on Windows 11. The interface includes a menu bar (File, View, Cache, Debug, Watch, Help), a toolbar, and several windows for RegistersView, CodeView, OutputView, and WatchView.

**RegistersView** (Registers View):

- General Purpose: Floating Point
- Hexadecimal
- Unsigned Decimal
- Signed Decimal

Register	Value
R0	:00001070
R1	:00001074
R2	:00001078
R3	:0000107c
R4	:00001080
R5	:00000002
R6	:00000000
R7	:00000021
R8	:00000068
R9	:00001078
R10 (s1)	:00000000
R11 (sp)	:00000000
R12 (ip)	:00000000
R13 (sp)	:00011400
R14 (lr)	:00000000
R15 (pc)	:00011400

**CodeView** (Code View):

```
.DATA
00001070:0000002C A : .WORD 44
00001074:00000002 B : .WORD 2
00001078:00000021 C : .WORD 33
0000107C:00000000 D : .WORD 0
00001080:00000000 E : .WORD 0

.TEXT
00001000:E59P0054 LDR r0,-A
00001004:E59P1054 LDR r1,-B
00001008:E59P2054 LDR r2,-C
0000100C:E59P3054 LDR r3,-D
00001010:E59P4054 LDR r4,-R
00001014:E59P5000 LDR r5,[r0]
00001018:E5916000 LDR r6,[r1]
0000101C:E5927000 LDR r7,[r2]
00001020:E59P903C LDR r9,-C
00001024:E1550006 CMP r5,r6
//IF A==B
00001028:0A000002 BEQ L1
0000102C:E1560007 CMP r5,r7
//Else if B==C
00001030:A0000003 BEQ L2
00001034:EA000005 B L3
//Else
L1:
00001038:E0858006 ADD r8,r5,r6
//C=A+B
```

**OutputView** (Output View):

```
Console stdIn/stdOut/stdErr
```

**WatchView** (Watch View):

Activate Windows  
Go to Settings to activate Windows.

### **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:Naman Choudhary

Name:Naman Choudhary

SRN:PES2UG20CS209

Section: D

Date:01/02/2022