

# Microprocessor and Computer Architecture

UE20CS252

4th Semester, Academic Year 2021-22

Date:

Name: Vishwa Mehul Mehta	SRN: PES2UG20CS389	Section: F
-----------------------------	-----------------------	---------------

Week# 4 Program Number: 1

Title of the Program

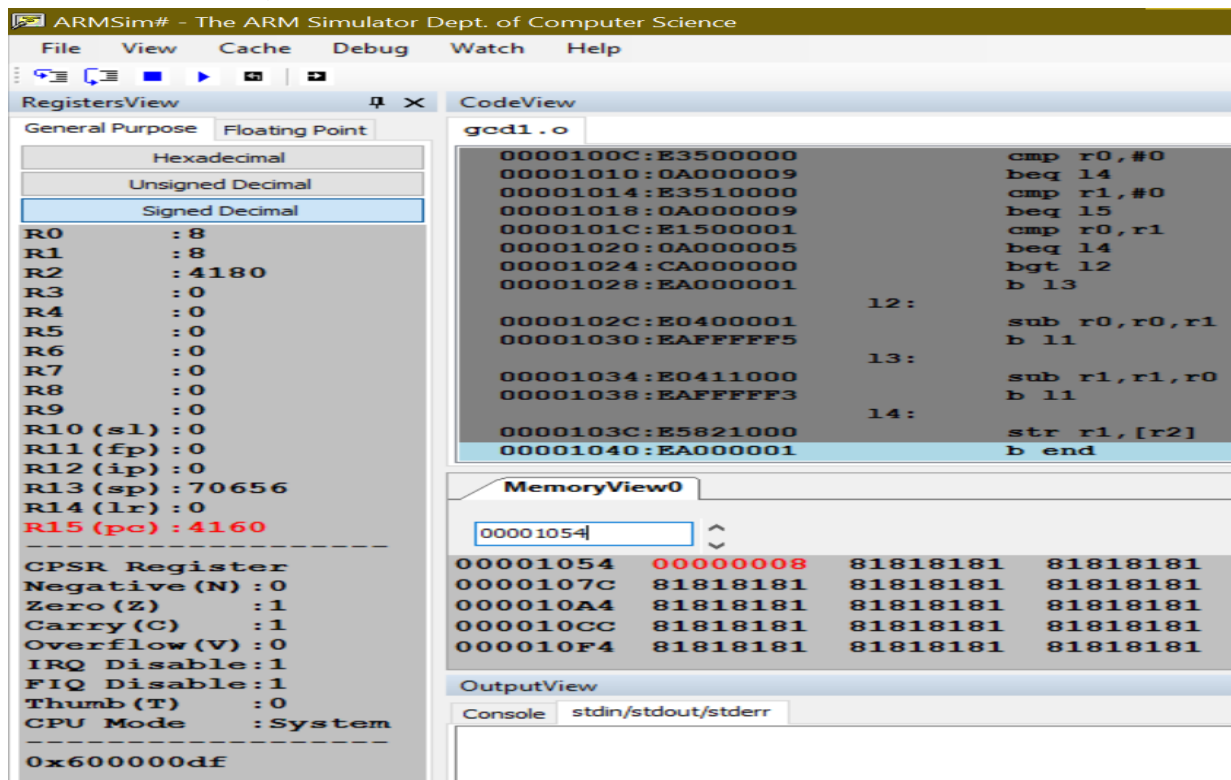
**Write a program in ARM7TDMI-ISA to find GCD of two numbers.**

**a. Assume operands to be in the CPU registers**

## 1. ARM Assembly Code(1)

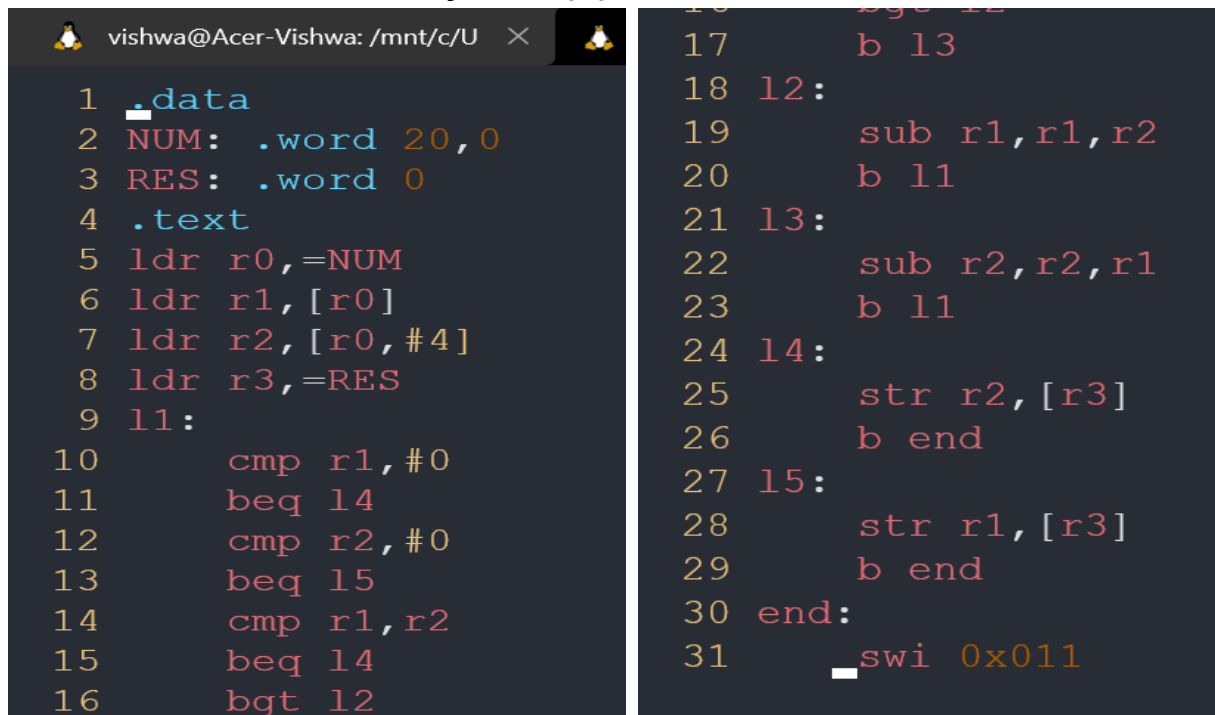
```
vishwa@Acer-Vishwa: /mnt/c/U ×  
1 .data  
2 RES: .word 0  
3 .text  
4 mov r0, #56  
5 mov r1, #16  
6 ldr r2, =RES  
7 l1:  
8     cmp r0, #0  
9     beq l4  
10    cmp r1, #0  
11    beq l5  
12    cmp r0, r1  
13    beq l4  
14    bgt l2  
15    b l3  
16 l2:  
17     sub r0, r0, r1  
18     b l1  
19 l3:  
20     sub r1, r1, r0  
21     b l1  
22 l4:  
23     str r1, [r2]  
24     b end  
25 l5:  
26     str r0, [r2]  
27     b end  
28 end:  
29     swi 0x011
```

## 2. Output Screen Shot (1)



b. Assume operands in the memory locations.

### 1. ARM Assembly Code(1)



## 2. Output Screen Shot (1)

ARMSim# - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose Floating Point

Hexadecimal  
Unsigned Decimal  
Signed Decimal

R0 : 4188  
R1 : 20  
R2 : 0  
R3 : 4196  
R4 : 0  
R5 : 0  
R6 : 0  
R7 : 0  
R8 : 0  
R9 : 0  
R10 (s1) : 0  
R11 (fp) : 0  
R12 (ip) : 0  
R13 (sp) : 70656  
R14 (lr) : 0  
**R15 (pc) : 4172**

-----  
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

CodeView

gcd2.o

```
0000102C:EA000001      b 13
00001030:E0411002      12:      sub r1,r1,r2
00001034:EAffFFFF5      13:      b 11
00001038:E0422001      13:      sub r2,r2,r1
0000103C:EAffFFFF3      14:      b 11
00001040:E5832000      14:      str r2,[r3]
00001044:EA000001      15:      b end
00001048:E5831000      15:      str r1,[r3]
0000104C:EAffFFFFF      b end
end:
00001050:EF000011      swi 0x011
      :00000000
      :00000008
```

MemoryView0

00001064

00001064	00000014	81818181	81818181
0000108C	81818181	81818181	81818181
000010B4	81818181	81818181	81818181
000010DC	81818181	81818181	81818181
00001104	81818181	81818181	81818181

OutputView

Console stdin/stdout/stderr

# Microprocessor and Computer Architecture

UE20CS252

4th Semester, Academic Year 2021-22

Date:

Name: Vishwa Mehul Mehta	SRN: PES2UG20CS389	Section: F
-----------------------------	-----------------------	---------------

Week# \_\_\_\_4\_\_\_\_

Program Number: \_\_\_\_2\_\_\_\_

Title of the Program

**Write a program in ARM7TDMI-ISA to find the sum of N data items at alternate [ odd or even positions] locations in the memory. Store the result in the memory location.**

**a. Use Pre-indexing addressing mode**

**Odd:**

## 1. ARM Assembly Code(1)

```
vishwa@Acer-Vishwa: /mnt/c/U  X  vishwa@Acer-Vishwa: /mnt/c/U
1  .data
2  NUM: .word 10,20,30,40,50
3  RES: .word 0
4  .text
5  ldr r0,=NUM
6  ldr r1,=RES
7  mov r2,#0 //sum
8  mov r3,#0 //step
9  mov r4,#1 //counter
10 ll:
11     ldr r5,[r0,r3]
12     add r2,r2,r5
13     add r3,r3,#8
14     add r4,r4,#2
15     cmp r4,#7
16     bne ll
17 str r2,[r1]
18 swi 0x011
```

## 2. Output Screen Shot (1)

The screenshot displays the ARMSim# - The ARM Simulator interface. The top menu bar includes File, View, Cache, Debug, Watch, and Help. The main window is divided into four panes:

- RegistersView:** Shows the state of ARM registers. The 'Signed Decimal' tab is selected. R0 is 4156, R1 is 4176, R2 is 90, R3 is 24, R4 is 7, R5 is 50, R6 is 0, R7 is 0, R8 is 0, R9 is 0, R10 (s1) is 0, R11 (fp) is 0, R12 (ip) is 0, R13 (sp) is 70656, R14 (lr) is 0, and R15 (pc) is 4144. The CPSR Register shows Negative (N) as 0, Zero (Z) as 1, Carry (C) as 1, Overflow (V) as 0, IRQ Disable as 1, FIQ Disable as 1, Thumb (T) as 0, and CPU Mode as System. The PC value is 0x600000df.
- CodeView:** Displays the assembly code for 'oddsum\_pre.o'. The code includes instructions for loading NUM and RES, moving values to R2, R3, and R4, a loop labeled 'l1' that adds R3 to R2 and increments R4, a comparison of R4 with 7, a branch if not equal to 'l1', storing R2 to [R1], and finally a software interrupt (swi 0x011).
- MemoryView0:** Shows memory contents starting from address 00001050. The value at 00001050 is 0000005A, and the value at 00001078 is 81818181. The memory view shows a pattern of 81818181 values.
- OutputView:** Shows the console output, which is currently empty.

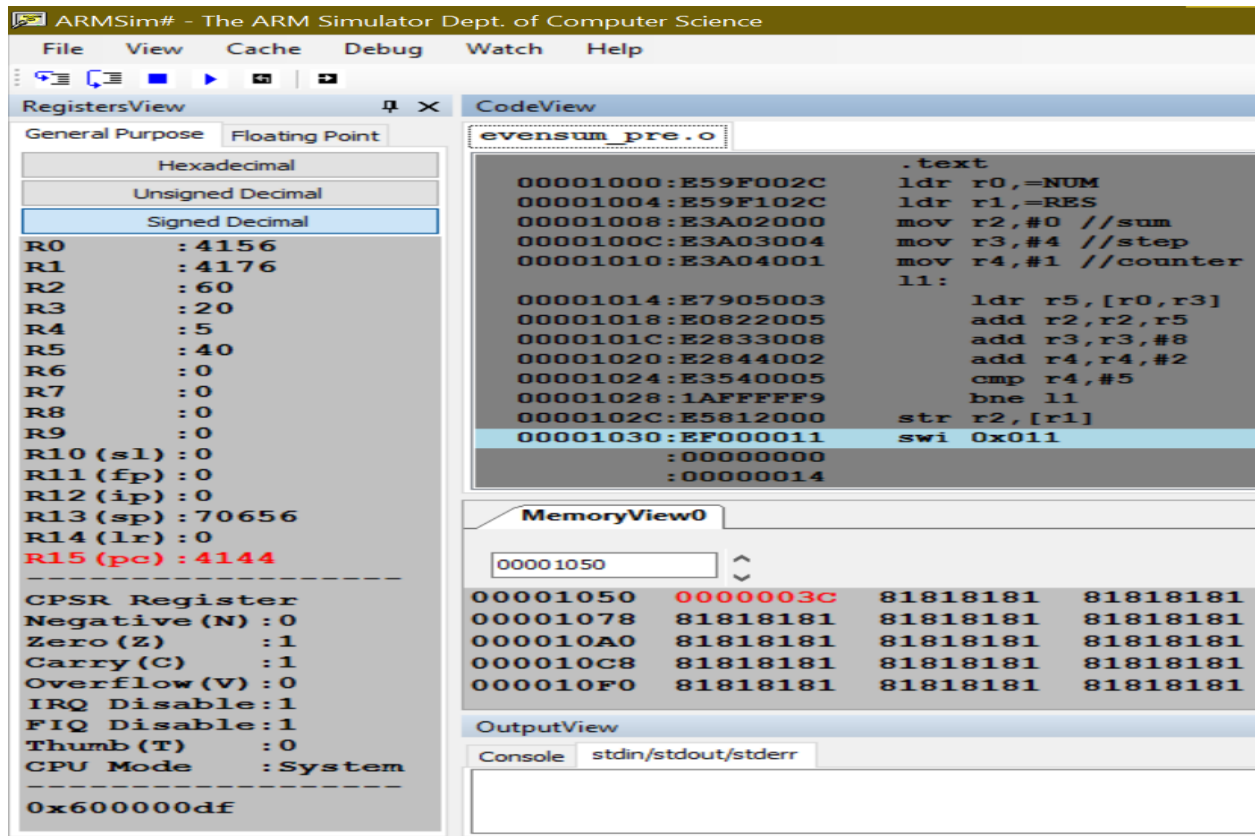
Even:

## 1. ARM Assembly Code(1)

```
vishwa@Acer-Vishwa: /mnt/c/U × vishwa@Acer-Vishwa: /mnt/c/U
```

```
1  .data
2  NUM: .word 10,20,30,40,50
3  RES: .word 0
4  .text
5  ldr r0,=NUM
6  ldr r1,=RES
7  mov r2,#0 //sum
8  mov r3,#4 //step
9  mov r4,#1 //counter
10 l1:
11     ldr r5,[r0,r3]
12     add r2,r2,r5
13     add r3,r3,#8
14     add r4,r4,#2
15     cmp r4,#5
16     bne l1
17 str r2,[r1]
18 swi 0x011
```

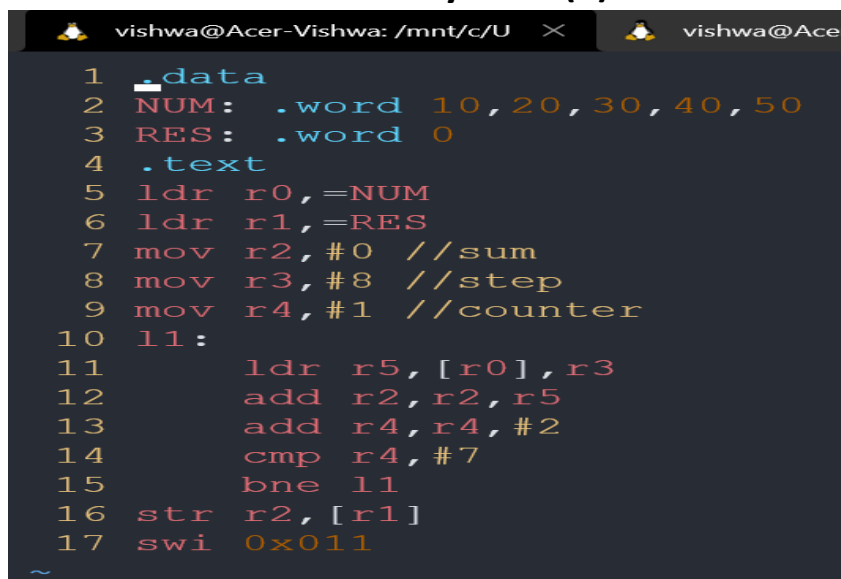
## 2. Output Screen Shot (1)



### b. Use Post- Indexing addressing mode

Odd:

## 1. ARM Assembly Code(1)



## 2. Output Screen Shot (1)

The screenshot displays the ARMSim# - The ARM Simulator interface. The top menu bar includes File, View, Cache, Debug, Watch, and Help. The main window is divided into four panes:

- RegistersView:** Shows the state of 16 registers (R0-R15) and the CPSR register. R0 is 4176, R1 is 4172, R2 is 90, R3 is 8, R4 is 7, R5 is 50, R6 is 0, R7 is 0, R8 is 0, R9 is 0, R10 (s1) is 0, R11 (fp) is 0, R12 (ip) is 0, R13 (sp) is 70656, R14 (lr) is 0, and R15 (pc) is 4140. The CPSR register shows Negative (N): 0, Zero (Z): 1, Carry (C): 1, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, and CPU Mode: System. The CPSR value is 0x600000df.
- CodeView:** Displays the assembly code for the file oddsum\_post.o. The code includes instructions like ldr, mov, add, cmp, bne, str, and swi. The current instruction being executed is swi 0x011 at address 0000102C.
- MemoryView0:** Shows the memory contents at address 0000104C. The value is 0000005A. The memory is organized into a table with columns for address, value, and other data.
- OutputView:** Shows the console output, which is currently empty.

Even:

## 1. ARM Assembly Code(1)

```
vishwa@Acer-Vishwa: /mnt/c/U  x  vishwa@Ace

1  .data
2  NUM: .word 10,20,30,40,50
3  RES: .word 0
4  .text
5  ldr r0,=NUM
6  ldr r1,=RES
7  mov r2,#0 //sum
8  mov r3,#8 //step
9  ldr r5,[r0],#4
10 mov r4,#1 //counter
11 l1:
12     ldr r5,[r0],r3
13     add r2,r2,r5
14     add r4,r4,#2
15     cmp r4,#5
16     bne l1
17 str r2,[r1]
18 swi 0x011
```

## 2. Output Screen Shot (1)

ARMSim# - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose Floating Point

Hexadecimal  
Unsigned Decimal  
Signed Decimal

R0 : 4176  
R1 : 4176  
R2 : 60  
R3 : 8  
R4 : 5  
R5 : 40  
R6 : 0  
R7 : 0  
R8 : 0  
R9 : 0  
R10 (s1) : 0  
R11 (fp) : 0  
R12 (ip) : 0  
R13 (sp) : 70656  
R14 (lr) : 0  
**R15 (pc) : 4144**

-----  
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

CodeView

evensum\_post.o

```
.text
00001000:E59F002C    ldr r0,=NUM
00001004:E59F102C    ldr r1,=RES
00001008:E3A02000    mov r2,#0 //sum
0000100C:E3A03008    mov r3,#8 //step
00001010:E4905004    ldr r5,[r0],#4
00001014:E3A04001    mov r4,#1 //counter
                                l1:
00001018:E6905003                ldr r5,[r0],r3
0000101C:E0822005                add r2,r2,r5
00001020:E2844002                add r4,r4,#2
00001024:E3540005                cmp r4,#5
00001028:1AFFFFFA                bne l1
0000102C:E5812000                str r2,[r1]
00001030:EF000011                swi 0x011
                                :00000000
                                :00000014
```

MemoryView0

00001050| ^

00001050	0000003C	81818181	81818181
00001078	81818181	81818181	81818181
000010A0	81818181	81818181	81818181
000010C8	81818181	81818181	81818181
000010F0	81818181	81818181	81818181

OutputView

Console stdin/stdout/stderr



**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: Vishwa Mehul Mehta

Name: Vishwa Mehul Mehta

SRN: PES2UG20CS389

Section: F

Date: 07/02/2022