

# Microprocessor and Computer Architecture

UE20CS252

4th Semester, Academic Year 2021-22

Date: 07/02/2022

Name: VISHWAS M	SRN: PES2UG20CS390	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**

**i)**

**MOV R1,#86**

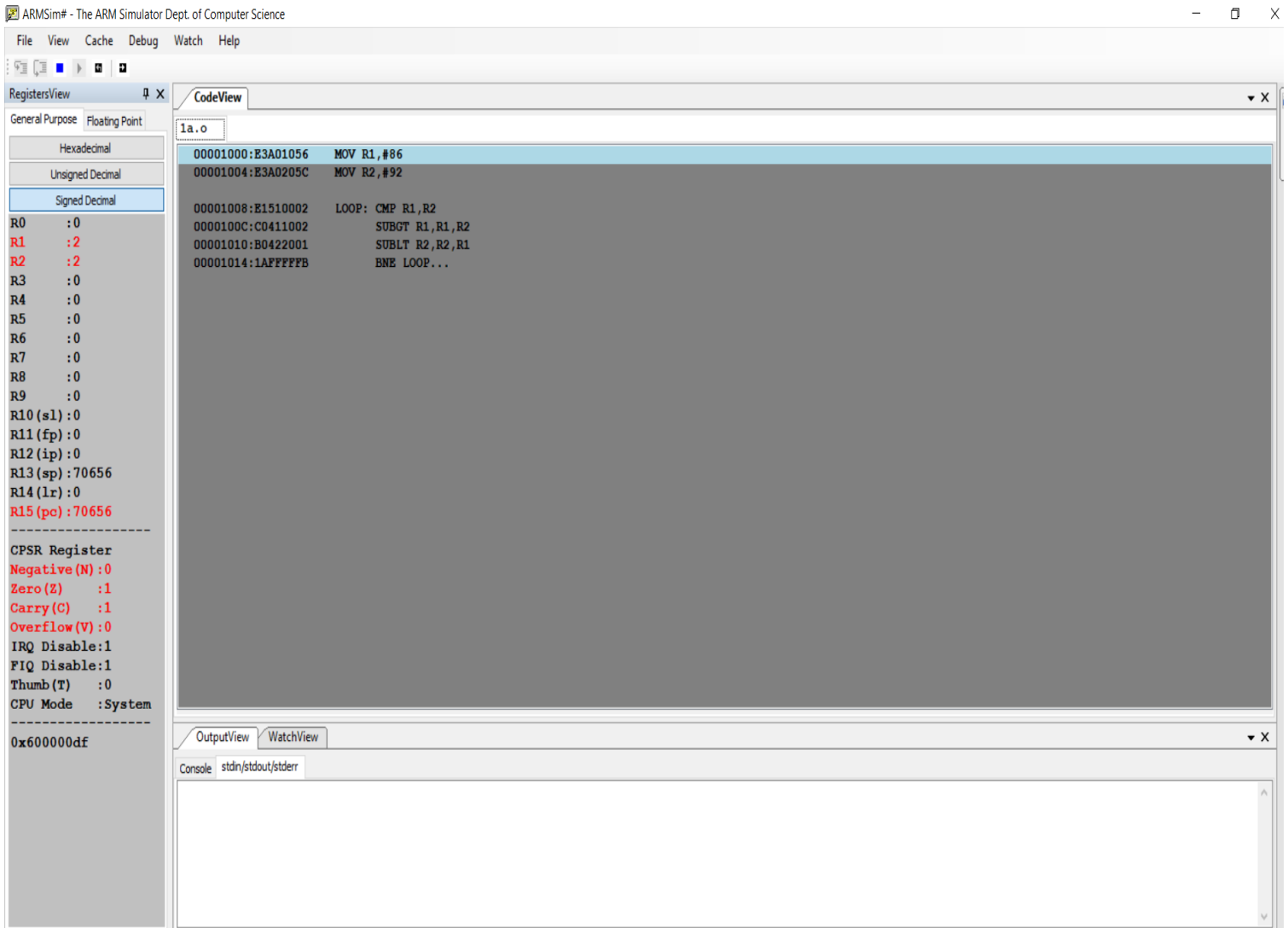
**MOV R2,#92**

**LOOP: CMP R1,R2**

**SUBGT R1,R1,R2**

**SUBLT R2,R2,R1**

**BNE LOOP**



ii)

**MOV R1,#86**

**MOV R2,#93**

**LOOP: CMP R1,R2**

**SUBGT R1,R1,R2**

**SUBLT R2,R2,R1**

**BNE LOOP**

ARMSim# - The ARM Simulator Dept. of Computer Science

FileViewCacheDebugWatchHelp

RegistersView

General PurposeFloating Point

HexadecimalUnsigned DecimalSigned Decimal

R0:0

R1:1

R2:1

R3:0

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):70656

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

1a.o

00001000:E3A01056MOV R1,#86

00001004:E3A0205DMOV R2,#93

00001008:E1510002LOOP: CMP R1,R2

0000100C:C0411002SUBGT R1,R1,R2

00001010:B0422001SUBLT R2,R2,R1

00001014:1AFFFFFBENE LOOP...

OutputViewWatchView

Consolestdin/stdout/stderr

**b. Assume operands in the memory locations.**

**i)**

**.DATA**

**A: .WORD 8,7**

**GCD: .WORD 0**

**.TEXT**

**LDR R0,=A**

**LDR R3,[R0]**

**ADD R0,R0,#4**

**LDR R4,[R0]**

**LDR R1,=GCD**

**LOOP: CMP R3,R4**

**SUBGT R3,R3,R4**

**SUBLT R4,R4,R3**

**BNE LOOP**

**STR R3,[R1]**

**SWI 0X011**

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 : 0  
R1 : 0  
R2 : 0  
R3 : 1  
R4 : 1  
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) : 70656

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

god\_register.o

```
//TO FIND SUM OF N DATA ITEMS IN THE MEMORY.STORE THE RESULT
//USING PRE-INDEXING ADDRESSING MODE

.DATA
00001034:00000008 A: .WORD 8,7
:00000007
0000103C:00000000 GCD: .WORD 0

.TEXT
00001000:E59F0024 LDR R0,=A
00001004:E5903000 LDR R3,[R0]
00001008:E2800004 ADD R0,R0,#4
0000100C:E5904000 LDR R4,[R0]
00001010:E59F1018 LDR R1,=GCD

00001014:E1530004 LOOP: CMP R3,R4
00001018:C0433004 SUBGT R3,R3,R4
0000101C:B0444003 SUBLT R4,R4,R3
00001020:1AFFFFF8 BNE LOOP
00001024:E5813000 STR R3,[R1]
00001028:EF000011 SWI 0X011

0000102C:00000000 :00000008
```

MemoryView0

0000103C

Word Size  
8Bt 16Bt 32Bt

0000103C	00000001	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001060	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001084	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010A8	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010CC	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010F0	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001114	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001138	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
0000115C	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001180	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000011A4	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000011C8	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181

OutputView WatchView

Console stdin/stdout/stderr

ii)

.DATA

A: .WORD 8,7

GCD: .WORD 0

.TEXT

LDR R0,=A

LDR R3,[R0]

ADD R0,R0,#4

LDR R4,[R0]

LDR R1,=GCD

LOOP: CMP R3,R4

SUBGT R3,R3,R4

SUBLT R4,R4,R3

BNE LOOP

STR R3,[R1]

SWI 0X011

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 : 0  
R1 : 0  
R2 : 0  
R3 : 7  
R4 : 7  
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) : 70656

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

gdc\_register.o

```
//TO FIND SUM OF N DATA ITEMS IN THE MEMORY.STORE THE RESULT  
//USING PRE-INDEXING ADDRESSING MODE  
  
.DATA  
00001034:00000007 A: .WORD 7,7  
:00000007  
0000103C:00000000 GCD: .WORD 0  
  
.TEXT  
00001000:E59F0024 LDR R0,=A  
00001004:E5903000 LDR R3,[R0]  
00001008:E2800004 ADD R0,R0,#4  
0000100C:E5904000 LDR R4,[R0]  
00001010:E59F1018 LDR R1,=GCD  
  
00001014:E1530004 LOOP: CMP R3,R4  
00001018:C0433004 SUBGT R3,R3,R4  
0000101C:B0444003 SUBLT R4,R4,R3  
00001020:1AFFFFFB BNE LOOP  
00001024:E5813000 STR R3,[R1]  
00001028:EF000011 SWI 0X011  
  
0000102C:00000000 :00000008
```

MemoryView0

Word Size 8Br 16Br 32Br

0000103C	00000007	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001060	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001084	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010A8	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010CC	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010F0	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001114	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001138	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
0000115C	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001180	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000011A4	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000011C8	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181

OutputView WatchView

Console stdin/stdout/stderr

# **Microprocessor and Computer Architecture**

**UE20CS252**

**4th Semester, Academic Year 2021-22**

Date: 07/02/2022

Name: VISHWAS M	SRN: PES2UG20CS390	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**

## i)For odd indices:

ARMSim# - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView CodeView

General Purpose Floating Point

Hexadecimal  
Unsigned Decimal  
Signed Decimal

R0 :0  
R1 :0  
R2 :44  
R3 :4172  
R4 :23  
R5 :16  
R6 :3  
R7 :0  
R8 :0  
R9 :0  
R10 (s1) :0  
R11 (fp) :0  
R12 (ip) :0  
R13 (sp) :70656  
R14 (lr) :0  
R15 (pc) :70656

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

0x60000df

pre\_index\_sum\_of\_alterate\_numbers.o

```
//TO FIND SUM OF N DATA ITEMS IN THE MEMORY.STORE THE RESULT
//USING PRE-INDEXING ADDRESSING MODE

.DATA
0000103C:00000015 A: .WORD 21,22,23,24
:00000016
:00000017
:00000018
0000104C:00000000 SUM: .WORD 0

.TEXT
00001000:E3A02000 MOV R2,#0
00001004:E59F1028 LDR R1,=A
00001008:E59F3028 LDR R3,=SUM
0000100C:E3A05000 MOV R5,#0 //FOR EVEN:#4
00001010:E3A06001 MOV R6,#1

00001014:E7914005 LOOP: LDR R4,[R1,R5]
00001018:E0822004 ADD R2,R2,R4
0000101C:E2855008 ADD R5,R5,#8
00001020:E2866001 ADD R6,R6,#1
00001024:E3560003 CMP R6,#3
00001028:1AFFFFF9 BNE LOOP
0000102C:E5832000 STR R2,[R3]
00001030:EF000011 SWI 0X011
00001034:00000000 .END...
:00000010
```

OutputView WatchView

Console stdin/stdout/stderr



## ii) For even indexes:

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 : 0  
R1 : 0  
R2 : 46  
R3 : 4172  
R4 : 24  
R5 : 20  
R6 : 3  
R7 : 0  
R8 : 0  
R9 : 0  
R10 (s1) : 0  
R11 (fp) : 0  
R12 (ip) : 0  
R13 (sp) : 70656  
R14 (lr) : 0  
R15 (pc) : 70656

-----  
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

pre\_index\_sum\_of\_alternate\_numbers.o

```
//TO FIND SUM OF N DATA ITEMS IN THE MEMORY.STORE THE RESULT
//USING PRE-INDEXING ADDRESSING MODE

.DATA
0000103C:00000015 A: .WORD 21,22,23,24
:00000016
:00000017
:00000018
0000104C:00000000 SUM: .WORD 0

.TEXT
00001000:E3A02000 MOV R2,#0
00001004:E59F1028 LDR R1, =A
00001008:E59F3028 LDR R3, =SUM
0000100C:E3A05004 MOV R5,#4 //FOR EVEN:#0
00001010:E3A06001 MOV R6,#1

00001014:E7914005 LOOP: LDR R4,[R1,R5]
00001018:E0822004 ADD R2,R2,R4
0000101C:E2855008 ADD R5,R5,#8
00001020:E2866001 ADD R6,R6,#1
00001024:E3560003 CMP R6,#3
00001028:1AFFFFFF9 BNE LOOP
0000102C:E5832000 STR R2,[R3]
00001030:EF000011 SWI 0X011
00001034:00000000 .END...
:00000010
```

OutputView WatchView

Console stdin/stdout/stderr

## b. Use Post- Indexing addressing mode

### i)For odd indices:

ARMSim# - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView CodeView

General Purpose Floating Point

Hexadecimal  
Unsigned Decimal  
Signed Decimal

R0 :0  
R1 :0  
R2 :44  
R3 :4176  
R4 :23  
R5 :16  
R6 :3  
R7 :0  
R8 :0  
R9 :0  
R10 (s1) :0  
R11 (fp) :0  
R12 (ip) :0  
R13 (sp) :70656  
R14 (lr) :0  
R15 (pc) :70656

-----  
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

post\_index\_sum\_of\_alterate\_numbers.o

```
//TO FIND SUM OF N DATA ITEMS IN THE MEMORY.STORE THE RESULT
//USING PRE-INDEXING ADDRESSING MODE

.DATA
00001040:00000015 A: .WORD 21,22,23,24
:00000016
:00000017
:00000018
00001050:00000000 SUM: .WORD 0

.TEXT
00001000:E3A02000 MOV R2,#0
00001004:E59F102C LDR R1, =A
00001008:E59F302C LDR R3, =SUM
0000100C:E3A05000 MOV R5,#0
00001010:E3A06001 MOV R6,#1

00001014:E6914005 LOOP: LDR R4,[R1],R5
00001018:E0822004 ADD R2,R2,R4
0000101C:E2855008 ADD R5,R5,#8
00001020:E0811005 ADD R1,R1,R5
00001024:E2866001 ADD R6,R6,#1
00001028:E3560003 CMP R6,#3
0000102C:1AFFFFF8 BNE LOOP
00001030:E5832000 STR R2,[R3]
00001034:EF000011 SWI 0X011
00001038:00000000 .END...
:00000010
```

OutputView WatchView

Console stdin/stdout/stderr

## ii) For even indices:

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 : 0  
R1 : 0  
R2 : 46  
R3 : 4180  
R4 : 24  
R5 : 16  
R6 : 3  
R7 : 0  
R8 : 0  
R9 : 0  
R10 (s1) : 0  
R11 (fp) : 0  
R12 (ip) : 0  
R13 (sp) : 70656  
R14 (lr) : 0  
R15 (pc) : 70656

-----  
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

post\_index\_sum\_of\_alterate\_numbers.o

```
//TO FIND SUM OF N DATA ITEMS IN THE MEMORY.STORE THE RESULT  
//USING PRE-INDEXING ADDRESSING MODE  
  
.DATA  
00001044:00000015 A: .WORD 21,22,23,24  
:00000016  
:00000017  
:00000018  
00001054:00000000 SUM: .WORD 0  
  
.TEXT  
00001000:E3A02000 MOV R2,#0  
00001004:E59F1030 LDR R1, =A  
00001008:E59F3030 LDR R3, =SUM  
0000100C:E3A05000 MOV R5,#0  
00001010:E2811004 ADD R1,R1,#4  
00001014:E3A06001 MOV R6,#1  
  
00001018:E6914005 LOOP: LDR R4,[R1],R5  
0000101C:E0822004 ADD R2,R2,R4  
00001020:E2855008 ADD R5,R5,#8  
00001024:E0811005 ADD R1,R1,R5  
00001028:E2866001 ADD R6,R6,#1  
0000102C:E3560003 CMP R6,#3  
00001030:1AFFFFF8 BNE LOOP  
00001034:E5832000 STR R2,[R3]  
00001038:EF000011 SWI 0X011  
0000103C:00000000 .END...  
:00000010
```

OutputView WatchView

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.

Name: VISHWAS M

SRN: PES2UG20CS390

Section: F

Date: 07/02/2022