

Microprocessor and Computer Architecture UE22CS251B
4th Semester, Academic Year 2023-2024

Date:

Name: Ankith Gowda B S	SRN: PES2UG22CS077	Section B
------------------------	-----------------------	--------------

Week# 1 Program Number: 1

Title of the Program

Write an ALP to add if the numbers are equal, otherwise subtract them.

I.ARM Assembly Code SOLUTION:

@NOTEQUAL CODE

.text

MOV R0,#20

MOV R1,#30

CMP R0,R1

BEQ L1

SUB R2,R0,R1

B L2

L1:ADD R2,R0,R1

```
L2:SWI 0X11
```

```
.end
```

```
@EQUAL CODE
```

```
.text
```

```
MOV R0,#30
```

```
MOV R1,#30
```

```
CMP R0,R1
```

```
BEQ L1
```

```
SUB R2,R0,R1
```

```
B L2
```

```
L1:ADD R2,R0,R1
```

```
L2:SWI 0X11
```

```
.end
```

II. Output Screen Shot (Two)

The output should be verified for both equal and
nor equal values SOLUTION:

NOT EQUAL

RegistersView

2

ARMv8-M

General Purpose

Run

Waiting Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0: 00000014

R1: 0000001e

R2: ffffffff

R3: 00000000

R4: 00000000

R5: 00000000

R6: 00000000

R7: 00000000

R8: 00000000

R9: 00000000

R10 (s1): 00000000

R11 (fp): 00000000

R12 (ip): 00000000

R13 (sp): 00000000

R14 (lr): 00001020

R15 (pc): 00000000

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: Supervisor

0x800000d3

#NOTEQUAL

.text

00001000: 23A00014 MOV R0, #20

00001004: 23A0101E MOV R1, #30

00001008: E1500001 CMP R0, R1

0000100C: 0A000001 BEQ L1

00001010: E0402001 SUB R2, R0, R1

00001014: EA000000 B L3

00001018: E0802001 L1: ADD R2, R0, R1

0000101C: EF000011 L2: SWI 0x11

.end

EQUAL:

RegistersView

2

ARMv8-M

General Purpose

Run

Waiting Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0: 0000001e

R1: 0000001e

R2: 0000003e

R3: 00000000

R4: 00000000

R5: 00000000

R6: 00000000

R7: 00000000

R8: 00000000

R9: 00000000

R10 (s1): 00000000

R11 (fp): 00000000

R12 (ip): 00000000

R13 (sp): 00000000

R14 (lr): 00001020

R15 (pc): 00000000

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: Supervisor

0x800000d3

#NOTEQUAL

.text

00001000: 23A0001E MOV R0, #30

00001004: 23A0101E MOV R1, #30

00001008: E1500001 CMP R0, R1

0000100C: 0A000001 BEQ L1

00001010: E0402001 SUB R2, R0, R1

00001014: EA000000 B L3

00001018: E0802001 L1: ADD R2, R0, R1

0000101C: EF000011 L2: SWI 0x11

.end

Microprocessor and Computer Architecture UE22CS251B
4th Semester, Academic Year 2023-2024

Date:

Name: Ankith Gowda B S	SRN: PES2UG22CS077	Section B
------------------------	-----------------------	--------------

Week# 1 Program Number: 2

Title of the Program

Write an ALP using ARM instruction set to check if a number stored in a register is even or odd. If even, store 00 in R0, else store FF in R0

I. ARM Assembly Code SOLUTION:

```
@FOR EVEN
.text
MOV R0,#8
AND R1,R0,#1 @AND OPERATION OF R0 AND 1 IN R1 TO
CHECK IF IT IS EVEN OR ODD
CMP R1,#0
BEQ EVEN
MOV R2,#1

B L2

EVEN:MOV R2,#0
L2:SWI 0X11
```

```
.end
```

```
@FOR ODD
```

```
.text
```

```
MOV R0,#9
```

```
AND R1,R0,#1 @AND OPERATION OF R0 AND 1 IN R1 TO  
CHECK IF IT IS EVEN OR ODD
```

```
CMP R1,#0
```

```
BEQ EVEN
```

```
MOV R2,#1
```

```
B L2
```

```
EVEN:MOV R2,#0
```

```
L2:SWI 0X11
```

```
.end
```

II. Output Screen Shot (Two)

The output should be verified for both even and odd numbers.

SOLUTION:

FOR EVEN:

```
Registers
General Purpose - Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 : 00000000
R1 : 00000000
R2 : 00000000
R3 : 00000000
R4 : 00000000
R5 : 00000000
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (sl) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00000000
R14 (lx) : 00001020
R15 (pc) : 00000000
-----
CPSR Register
Negative(N) : 0
Zero(Z) : 1
Carry(C) : 1
Overflow(O) : 0
IRQ Disable: 1
FIQ Disable: 1
Thumb(T) : 0
CPU Mode : Supervisor
0x500000d3

armv7a2.s
EVEN: EVEN
;test
00001000: 03a00000 MOV R0, #0
00001004: 02001001 AND R1, R0, #1 SAME OPERATION OF R0 AND 1 IN R1 TO CHECK IF IT IS EVEN OR ODD
00001008: 01010000 CMP R1, #0
0000100c: 0a000001 BEQ EVEN
00001010: 03a01001 MOV R2, #1
00001014: 0a000000 B L2
00001018: 03a02000 EVEN: MOV R2, #0
0000101c: 0f000011 L2: BGT NEXT
00001020: 40000000 ADD
```

FOR ODD:

```
Registers
General Purpose - Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 : 00000000
R1 : 00000001
R2 : 00000001
R3 : 00000000
R4 : 00000000
R5 : 00000000
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (sl) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00000000
R14 (lx) : 00001020
R15 (pc) : 00000000
-----
CPSR Register
Negative(N) : 0
Zero(Z) : 0
Carry(C) : 1
Overflow(O) : 0
IRQ Disable: 1
FIQ Disable: 1
Thumb(T) : 0
CPU Mode : Supervisor
0x200000d3

armv7a2.s
ODD: ODD
;test
00001000: 03a00000 MOV R0, #0
00001004: 02001001 AND R1, R0, #1 SAME OPERATION OF R0 AND 1 IN R1 TO CHECK IF IT IS EVEN OR ODD
00001008: 01010000 CMP R1, #0
0000100c: 0a000001 BEQ ODD
00001010: 03a01001 MOV R2, #1
00001014: 0a000000 B L2
00001018: 03a02000 ODD: MOV R2, #0
0000101c: 0f000011 L2: BGT NEXT
00001020: 40000000 ADD
```

**Microprocessor and Computer Architecture
UE22CS251B 4th Semester, Academic Year 2023-
2024**

Date:

Name: Ankith Gowda B S	SRN: PES2UG22CS077	Section B
------------------------	-----------------------	--------------

Week# 1

Program Number: 3

Title of the Program

Write a program to find the factorial of a given number.

I.ARM Assembly Code SOLUTION:

.text

MOV R0,#5

MOV R1,#1

L1:MUL R1,R0,R1 @FOR LOOP UNTIL R0 REACHES 1

SUB R0,R0,#1

CMP R0,#1

BNE L1

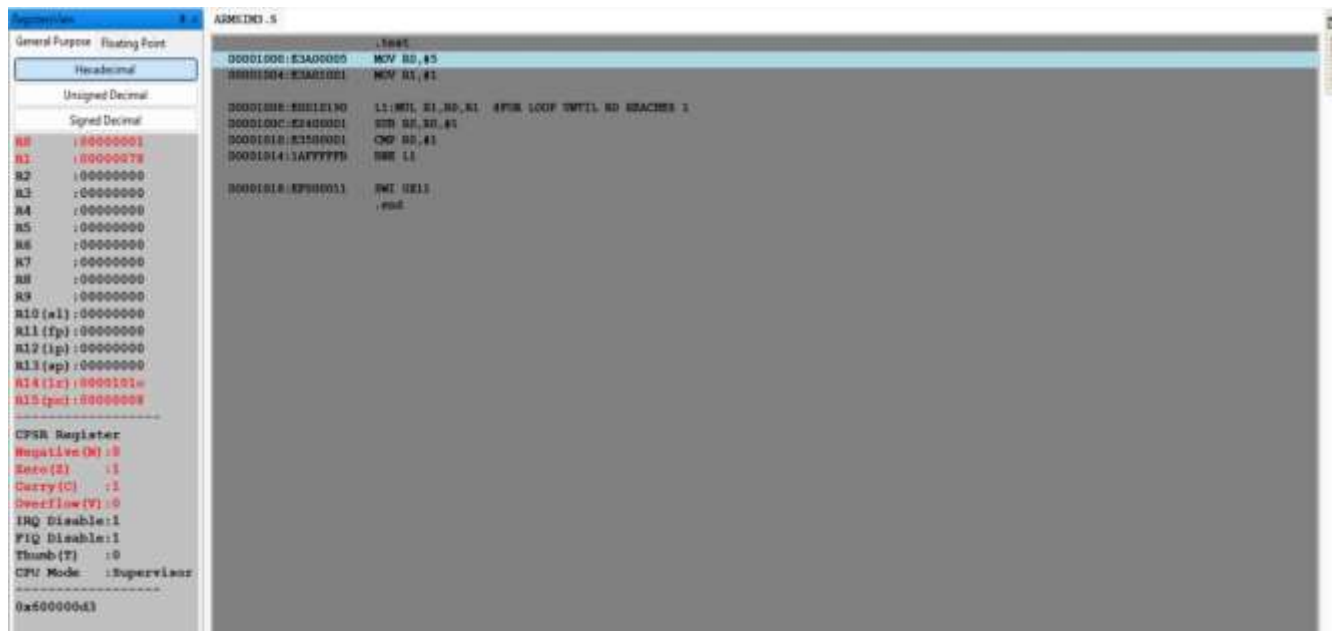
SWI 0X11

.end

II. Output Screen Shot (One)

The output should be verified for one number

SOLUTION:



**Microprocessor and Computer Architecture
UE22CS251B 4th Semester, Academic Year 2023-
2024**

Date: 19 – 01 -2024

Name: Ankith Gowda B S	SRN: PES2UG22CS077	Section B
------------------------	-----------------------	--------------

Week#____1_____

Program Number: ____4__

Title of the Program

Write a program to find GCD of two numbers.

I.ARM Assembly Code SOLUTION:

.text

MOV R0, #15 @15,15

MOV R1, #15 @10,20

GCD: CMP R0, R1

BEQ GCD_EQ

BGT GCD_G

BLT GCD_L

GCD_G: SUB R0, R0, R1

CMP R0, #1

BLT L2

B GCD

GCD_L: SUB R1, R1, R0

CMP R1, #1

BLT L2

B GCD

GCD_EQ: MOV R2,R1

B L

L2: MOV R2, #1

B L

L: SWI 0X11

.end

II. Output Screen Shot (Three)

The output should be verified for three cases

SOLUTION:

(1st Number== 2nd Number)

Register file: 00000000

General Purpose Register: Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0: 00000000
R1: 00000000
R2: 00000000
R3: 00000000
R4: 00000000
R5: 00000000
R6: 00000000
R7: 00000000
R8: 00000000
R9: 00000000
R10 (s1): 00000000
R11 (fp): 00000000
R12 (ip): 00000000
R13 (sp): 00000000
R14 (lr): 00000000
R15 (pc): 00000000

CPSR Register

Negative(N): 0
Zero(Z): 1
Carry(C): 1
Overflow(O): 0
IRQ Disable: 1
FIQ Disable: 1
Thumb(T): 0
CPU Mode: Supervisor
0x00000000

Assembly code:

```

00000000: E3A0000F MOV R0, #15
00000004: E3A0100F MOV R1, #15
00000008: E1500011 GCD: CMP R0, R1
0000000C: DAB00009 BGT GCD_RQ
00000010: DAB00000 BGT GCD_0
00000014: DAB00000 BGT GCD_1
00000018: E0400001 GCD_1: SUB R0, R0, R1
0000001C: E3500001 CMP R0, #1
00000020: DAB00004 BGT L3
00000024: DAF0000F B GCD
00000028: E0411000 GCD_L: SUB R1, R1, R0
0000002C: E3510001 CMP R1, #1
00000030: DAB10000 BGT L3
00000034: DAF0000F B GCD
00000038: E1A00001 GCD_RQ: MOV R2, R1
0000003C: DAB00001 B L
00000040: E1A00001 L3: MOV R2, #1
00000044: DAF0000F B L
00000048: EF000011 L: SWI 0x11
                                .end

```

(1st Number > 2nd Number)

Register file: 00000000

General Purpose Register: Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0: 00000000
R1: 00000000
R2: 00000000
R3: 00000000
R4: 00000000
R5: 00000000
R6: 00000000
R7: 00000000
R8: 00000000
R9: 00000000
R10 (s1): 00000000
R11 (fp): 00000000
R12 (ip): 00000000
R13 (sp): 00000000
R14 (lr): 00000000
R15 (pc): 00000000

CPSR Register

Negative(N): 0
Zero(Z): 1
Carry(C): 1
Overflow(O): 0
IRQ Disable: 1
FIQ Disable: 1
Thumb(T): 0
CPU Mode: Supervisor
0x00000000

Assembly code:

```

00000000: E3A0000F MOV R0, #15
00000004: E3A0100F MOV R1, #15
00000008: E1500011 GCD: CMP R0, R1
0000000C: DAB00009 BGT GCD_RQ
00000010: DAB00000 BGT GCD_0
00000014: DAB00000 BGT GCD_1
00000018: E0400001 GCD_1: SUB R0, R0, R1
0000001C: E3500001 CMP R0, #1
00000020: DAB00004 BGT L3
00000024: DAF0000F B GCD
00000028: E0411000 GCD_L: SUB R1, R1, R0
0000002C: E3510001 CMP R1, #1
00000030: DAB10000 BGT L3
00000034: DAF0000F B GCD
00000038: E1A00001 GCD_RQ: MOV R2, R1
0000003C: DAB00001 B L
00000040: E1A00001 L3: MOV R2, #1
00000044: DAF0000F B L
00000048: EF000011 L: SWI 0x11
                                .end

```

(1st Number < 2nd Number)

Registers

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 0

R2 : 0

R3 : 0

R4 : 0

R5 : 0

R6 : 0

R7 : 0

R8 : 0

R9 : 0

R10 (a1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 0

R14 (lr) : 4172

R15 (pc) : 0

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 : Supervisor

0x000000d3

armv7m.d

Write an ALP to find the GCD of two numbers (without using LDR and STR instructions). Both are registers, use only registers.

```

.text
00001000: E3A0000F  MOV R0, #15
00001004: E3A01014  MOV R1, #20
00001008: E1500001  GCD: CMP R0, R1
0000100C: DA000029  BEQ GCD_EQ
00001010: DA000000  SUB GCD, R0
00001014: DA000003  SUB GCD, R1
00001018: 09400001  GCD_L: SUB R0, R0, R1
0000101C: E3000001  CMP R0, #1
00001020: DA000006  SUB R2, R1
00001024: DAFFFFFF  S GCD
00001028: 00411000  GCD_L: SUB R1, R1, R0
0000102C: E3010001  CMP R1, #1
00001030: DA000003  SUB R2, R1
00001034: DAFFFFFF  S GCD
00001038: E1A02001  GCD_EQ: MOV R2, R1
0000103C: EAD00001  B L
00001040: E3A01001  LDR MOV R2, #1
00001044: DAFFFFFF  S L
00001048: E7F00011  LDR R0, R21

```

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:

Name: Ankith Gowda B S

SRN: PES2UG22CS077

Section: B

Date: 19-01-2024