

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nihal T M	SRN: PES2UG21CS333	Section:F
-----------------	--------------------	-----------

Week# 1 Program Number: 1

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. Code:

```
.text
```

```
LDR R0,=a
```

```
LDR R1,[r0]
```

```
AND R2,R1,#1
```

```
CMP R2, #0
```

```
BEQ even
```

```
BNE odd
```

```
even: MOV R0,#00
```

B END

odd: MOV R0, #255

B END

END:

.data

a:.word 10

II. Screen Shot:

a) when number is even

The screenshot shows a debugger window with two main panes. The left pane displays the state of the processor registers and the CPSR. The right pane shows the assembly code being debugged.

Left Pane (Registers and CPSR):

- Registers: R0 through R14 are shown with values 00000000. R15 (PC) is 00001028.
- 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.
- Memory: 0x600000df.

Right Pane (Assembly Code):

```
.text
00001000:E59F0020    LDR R0,=a
00001004:E5901000    LDR R1,[r0]
00001008:E2012001    AND R2,R1,#1
0000100C:E3520000    CMP R2, #0
00001010:0A000000    BEQ even
00001014:1A000001    BNE odd

00001018:E3A00000    even: MOV R0,#00
0000101C:EA000001    B END

00001020:E3A000FF    odd: MOV R0, #255
00001024:EAFF0000    B END

00001028:                END:

.data
0000102C:                a:.word 10
```

b) when number is odd

Floating Point | p4.s

Register	Value
R0	:000000ff
R1	:0000000b
R2	:00000001
R3	:00000000
R4	:00000000
R5	:00000000
R6	:00000000
R7	:00000000
R8	:0000005b
R9	:00000000
R10(sl)	:00000000
R11(fp)	:00000000
R12(ip)	:00000000
R13(sp)	:00005400
R14(lr)	:00000000
R15(pc)	:00011400

 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

```

.text
00001000:E59F0020    LDR R0,=a
00001004:E5901000    LDR R1,[r0]
00001008:E2012001    AND R2,R1,#1
0000100C:E3520000    CMP R2,#0
00001010:0A000000    BEQ even
00001014:1A000001    BNE odd

00001018:E3A00000    even: MOV R0,#00
0000101C:EA000001    B END

00001020:E3A000FF    odd: MOV R0,#255
00001024:EAF00000    B END

00001028:                END:

.data
0000102C:                a:.word 11
  
```

III. Output Table:

Input	Output
R1=10	R0=00
R1=11	R0=ff (255)

Week#____1_____

Program Number: ____2____

Title of the Program

**Write an ALP to compare the value of R0 and R1, add if R0
= R1, else subtract**

I. ARM Assembly Code:

.text

LDR R0,=a

LDR R1,=b

LDR R2,=c

LDR R3,[r0]

LDR R4,[r1]

CMP R3,R4

beq add

sub r5,r3,R4

STR r5,[r2]

B END

add:

add r5,r3,R4

STR r5,[r2]

END:

.data

a:.word 10

b:.word 6

c:.word 0

II. Output Screen Shot (Two)

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

a. when r0 and r1 are equal:

The screenshot shows a debugger window with two panes. The left pane displays the state of the processor registers and CPSR. The right pane displays the assembly code for the program.

Register Values:

Register	Value
R0	:0
R1	:0
R2	:4164
R3	:10
R4	:10
R5	:20
R6	:0
R7	:0
R8	:0
R9	:0
R10(sl)	:0
R11(fp)	:0
R12(ip)	:0
R13(sp)	:21504
R14(lr)	:0
R15(pc)	:70656

CPSR Register:

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

Assembly Code (p5.s):

```
.text
00001000:E59F0028 LDR R0,=a
00001004:E3A01D41 LDR R1,=b
00001008:E59F2024 LDR R2,=c
0000100C:E5903000 LDR R3,[r0]
00001010:E5914000 LDR R4,[r1]
00001014:E1530004 CMP R3,R4
00001018:0A000002 beq add
0000101C:E0435004 sub r5,r3,R4
00001020:E5825000 STR r5,[r2]
00001024:EA000001 B END
00001028: add:
00001028:E0835004 add r5,r3,R4
0000102C:E5825000 STR r5,[r2]
00001030: END:

.data
0000103C: a:.word 10
00001040: b:.word 10
00001044: c:.word 0
```

b. when r0 and r1 are not equal:

The screenshot shows a debugger window with the file 'p5.s' open. The left pane displays the state of 16 registers (R0-R15) and the CPSR register. The right pane shows the assembly code for the program.

Register	Value
R0	:4156
R1	:4160
R2	:4164
R3	:10
R4	:6
R5	:4
R6	:0
R7	:0
R8	:37440
R9	:0
R10(sl)	:0
R11(fp)	:0
R12(ip)	:0
R13(sp)	:21504
R14(lr)	:0
R15(pc)	:70656

Register	Flag/Status
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


```

.text
00001000:E59F0028    LDR R0,=a
00001004:E3A01D41    LDR R1,=b
00001008:E59F2024    LDR R2,=c
0000100C:E5903000    LDR R3,[r0]
00001010:E5914000    LDR R4,[r1]
00001014:E1530004    CMP R3,R4
00001018:0A000002    beq add
0000101C:E0435004    sub r5,r3,R4
00001020:E5825000    STR r5,[r2]
00001024:EA000001    B END
00001028:                add:
00001028:E0835004    add r5,r3,R4
0000102C:E5825000    STR r5,[r2]
00001030:                END:

.data
0000103C:                a:.word 10
00001040:                b:.word 6
00001044:                c:.word 0
  
```

III. Output Table:

Input	Output
R3=10, R4=10	R5=20
R3=10, R4=6	R5=4

Week#____1_____

Program Number: ____3____

Title of the Program

Based on the value of the number in R0, Write an ALP to store 1 in R1 if R0 is zero, Store 2 in R1 if R0 is positive, Store 3 in R1 if R0 is negative. (Program shown in class)

I. ARM Assembly Code:

```
.text
```

```
ldr r0,=a
```

```
ldr r2,[r0]
```

```
CMP r2,#0
```

```
BEQ zero
```

```
BPL plus
```

```
BMI negative
```

```
zero: mov r1,#1
```

```
B END
```

```
plus: mov r1,#2
```

```
B END
```

```
negative: mov r1,#3
```

```
B END
```

```
END:
```

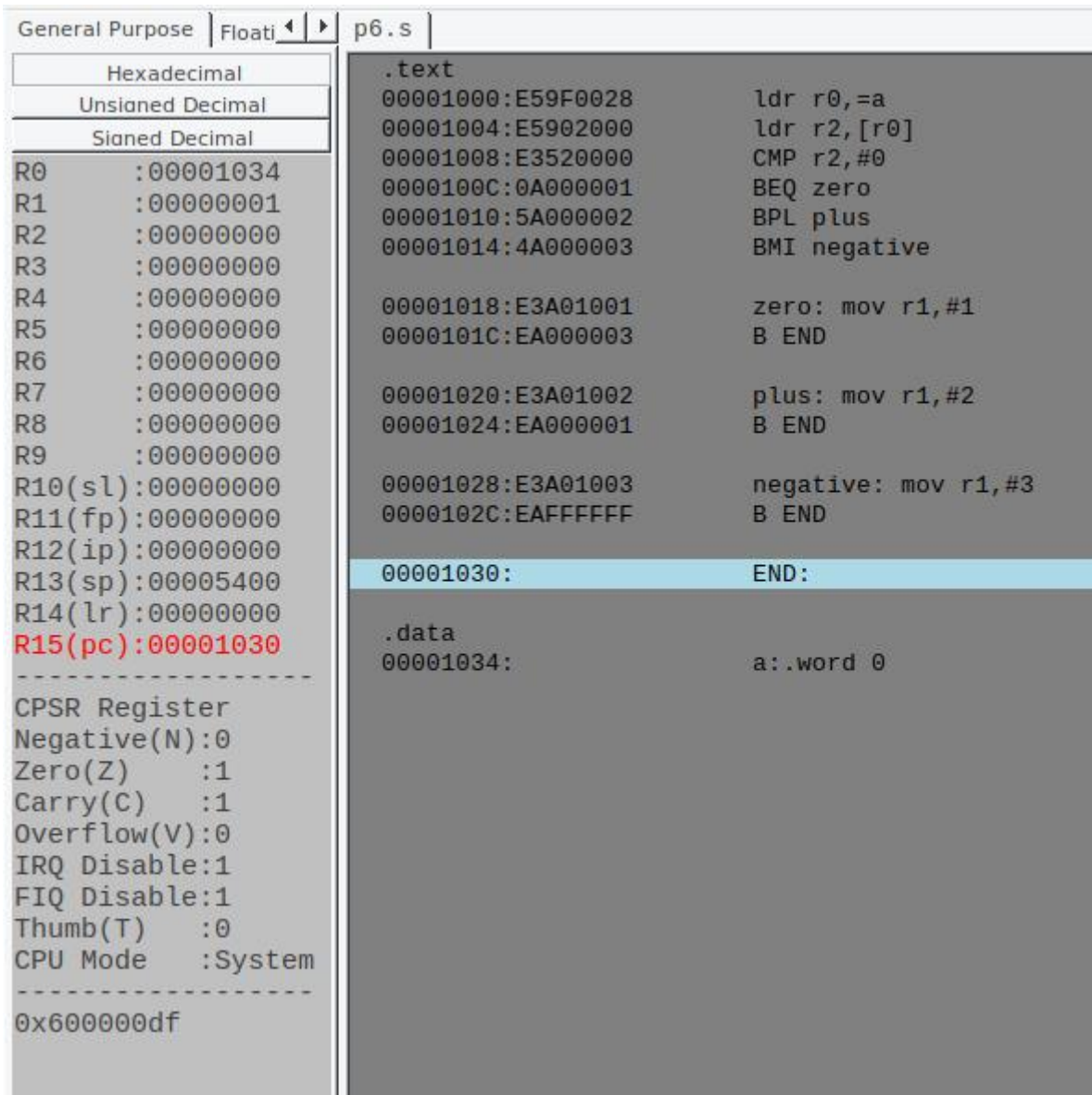
```
.data
```

```
a:.word 0
```

II. Output Screen Shot (Three)

The output should be verified for zero, positive and negative cases.

a. when the number is 0:



The screenshot shows a debugger window with the following components:

- General Purpose** tab selected, showing register values in Signed Decimal format.
- Floats** tab selected, showing floating-point register values in Signed Decimal format.
- p6.s** file loaded, showing assembly code.
- Assembly Code:**
 - `.text`
 - `00001000:E59F0028 ldr r0,=a`
 - `00001004:E5902000 ldr r2,[r0]`
 - `00001008:E3520000 CMP r2,#0`
 - `0000100C:0A000001 BEQ zero`
 - `00001010:5A000002 BPL plus`
 - `00001014:4A000003 BMI negative`
 -
 - `00001018:E3A01001 zero: mov r1,#1`
 - `0000101C:EA000003 B END`
 -
 - `00001020:E3A01002 plus: mov r1,#2`
 - `00001024:EA000001 B END`
 -
 - `00001028:E3A01003 negative: mov r1,#3`
 - `0000102C:EAffffff B END`
 -
 - `00001030: END:`
 -
 - `.data`
 - `00001034: a:.word 0`
- Registers:**
 - R0: 00001034
 - R1: 00000001
 - 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): 00005400
 - R14(lr): 00000000
 - R15(pc): 00001030**
- 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
- Memory:** 0x600000df

b. when number is positive:

Floating Point		p6.s
Hexadecimal		
Unsigned Decimal		
Signed Decimal		
R0	:4148	.text
R1	:2	00001000:E59F0028 ldr r0,=a
R2	:10	00001004:E5902000 ldr r2,[r0]
R3	:0	00001008:E3520000 CMP r2,#0
R4	:0	0000100C:0A000001 BEQ zero
R5	:0	00001010:5A000002 BPL plus
R6	:0	00001014:4A000003 BMI negative
R7	:0	
R8	:18	00001018:E3A01001 zero: mov r1,#1
R9	:0	0000101C:EA000003 B END
R10(sl)	:0	
R11(fp)	:0	00001020:E3A01002 plus: mov r1,#2
R12(ip)	:0	00001024:EA000001 B END
R13(sp)	:21504	
R14(lr)	:0	00001028:E3A01003 negative: mov r1,#3
R15(pc)	:70656	0000102C:EAF00000 B END
CPSR Register		00001030: END:
Negative(N):0		.data
Zero(Z):0		00001034: a:.word 10
Carry(C):1		
Overflow(V):0		
IRQ Disable:1		
FIQ Disable:1		
Thumb(T):0		
CPU Mode: System		
0x200000df		

c. when number is negative:

Floating Point		p6.s
Hexadecimal		
Unsigned Decimal		
Signed Decimal		
R0	:4148	.text
R1	:3	00001000:E59F0028 ldr r0,=a
R2	:-10	00001004:E5902000 ldr r2,[r0]
R3	:0	00001008:E3520000 CMP r2,#0
R4	:0	0000100C:0A000001 BEQ zero
R5	:0	00001010:5A000002 BPL plus
R6	:0	00001014:4A000003 BMI negative
R7	:0	
R8	:27	00001018:E3A01001 zero: mov r1,#1
R9	:0	0000101C:EA000003 B END
R10(sl)	:0	
R11(fp)	:0	00001020:E3A01002 plus: mov r1,#2
R12(ip)	:0	00001024:EA000001 B END
R13(sp)	:21504	
R14(lr)	:0	00001028:E3A01003 negative: mov r1,#3
R15(pc)	:70656	0000102C:EAF00000 B END
CPSR Register		00001030: END:
Negative(N):1		.data
Zero(Z):0		00001034: a:.word -10
Carry(C):1		
Overflow(V):0		
IRQ Disable:1		
FIQ Disable:1		
Thumb(T):0		
CPU Mode: System		
0xa00000df		

III.Output Table

Input	Output
R0=0	R1=1
R0=10	R1=2
R0=-10	R1=3

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:Nihal T M

SRN:PES2UG21CS333

Section: F

Date:16/01/2023