

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Name: Atul Anurag	SRN: PES2UG19CS075	Section: B
-----------------------------	------------------------------	----------------------

Date: 03-02-2021

WEEK#2

Program Number: 1

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.

I. ARM Assembly Code for each program

[a] Week2_Program1_PES2UG19CS075.s

```
1  .text
2  mov R0, #0    @case 0
3  cmp R0, #0
4
5  movEQ R1, #1
6  BEQ end
7
8  movMI R1, #3
9  BMI end
10
11 mov R1, #2
12
13 end:
14 SWI 0x011
15 .end
16
```

[a] Week2_Program1_PES2UG19CS075.s

```
1  .text
2  mov R0, #5    @case positive
3  cmp R0, #0
4
5  movEQ R1, #1
6  BEQ end
7
8  movMI R1, #3
9  BMI end
10
11 mov R1, #2
12
13 end:
14 SWI 0x011
15 .end
16
```

[a] Week2_Program1_PES2UG19CS075.s

```
1  .text
2  mov R0, #-5   @case negative
3  cmp R0, #0
4
5  movEQ R1, #1
6  BEQ end
7
8  movMI R1, #3
9  BMI end
10
11 mov R1, #2
12
13 end:
14 SWI 0x011
15 .end
16
```

II. Output Screen Shot

RegistersView

General Purpose | Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 00000000
R1 : 00000001
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) : 00011400
R14 (lr) : 00000000
R15 (pc) : 00011400

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

Week2_Program1_PES2UG19CS075.o

```
.text
00001000:E3A00000  mov R0, #0    @case 0
00001004:E3500000  cmp R0, #0
00001008:03A01001  movEQ R1, #1
0000100C:0A000002  BEQ end
00001010:43A01003  movMI R1, #3
00001014:4A000000  BMI end
00001018:E3A01002  mov R1, #2
0000101C:EF000011  end:
SWI 0x011
.end
```

OutputView | WatchView

Console | stdin/stdout/stderr

Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program1_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400
Execution ending, Instruction Count:16637 Elapsed Time:00:00:00.0482764
Instructions per second:344619

RegistersView

General Purpose | Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 00000005
R1 : 00000002
R2 : 00000000
R3 : 00000000
R4 : 00000000
R5 : 00000000
R6 : 00000000
R7 : 00000000
R8 : 00000012
R9 : 00000000
R10 (s1) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00011400
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

CodeView

Week2_Program1_PES2UG19CS075.o

```
.text
00001000:E3A00005  mov R0, #5    @case positive
00001004:E3500000  cmp R0, #0
00001008:03A01001  movEQ R1, #1
0000100C:0A000002  BEQ end
00001010:43A01003  movMI R1, #3
00001014:4A000000  BMI end
00001018:E3A01002  mov R1, #2
0000101C:EF000011  end:
SWI 0x011
.end
```

OutputView | WatchView

Console | stdin/stdout/stderr

Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program1_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400
Execution ending, Instruction Count:16640 Elapsed Time:00:00:00.0491532
Instructions per second:338533

RegistersView

General Purpose

Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 :fffffffb
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :0000001b
R9 :00000000
R10 (s1) :00000000
R11 (fp) :00000000
R12 (ip) :00000000
R13 (sp) :00011400
R14 (lr) :00000000
R15 (pc) :00011400

CPSR Register

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

0xa00000df

CodeView

Week2_Program1_PES2UG19CS075.o

```

.text
00001000:E3E00004  mov R0, #-5 @case negative
00001004:E3500000  cmp R0, #0
00001008:03A01001  movEQ R1, #1
0000100C:0A000002  BEQ end
00001010:43A01003  movMI R1, #3
00001014:4A000000  BMI end
00001018:E3A01002  mov R1, #2
end:
0000101C:EF000011  SWI 0x011
.end

```

OutputView

WatchView

Console

stdin/stdout/stderr

Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program1_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400
Execution ending, Instruction Count:16639 Elapsed Time:00:00:00.0576817
Instructions per second:288462

Program Number: 2

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

I. ARM Assembly Code for each program

[10] Week2_Program2_PES2UG19CS075.s

```
1  .text
2
3  mov r0, #3
4  mov r1, #3
5  cmp r0, r1    @equal case
6
7  bEQ L1
8
9  sub r2,r1,r0
10 b end
11
12 L1:
13 add r2,r1,r0
14
15 end:
16 swi 0x011
17 .end
```

[10] Week2_Program2_PES2UG19CS075.s

```
1  .text
2
3  mov r0, #3
4  mov r1, #5
5  cmp r0, r1    @unequal case
6
7  bEQ L1
8
9  sub r2,r1,r0
10 b end
11
12 L1:
13 add r2,r1,r0
14
15 end:
16 swi 0x011
17 .end
```

II. Output Screen Shot

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 :00000003

R1 :00000003

R2 :00000006

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) :00011400

R14 (lr) :00000000

R15 (pc) :00011400

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

Week2_Program2_PES2UG19CS075.o

.text

00001000:E3A00003 mov r0, #3

00001004:E3A01003 mov r1, #3

00001008:E1500001 cmp r0, r1 @equal case

0000100C:0A000001 bEQ L1

00001010:E0412000 sub r2,r1,r0

00001014:EA000000 b end

L1:

00001018:E0812000 add r2,r1,r0

end:

0000101C:EF000011 swi 0x011

.end...

OutputView WatchView

Console stdin/stdout/stderr

Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program2_PES2UG19CS075.s

Execution starting ...

Location 113FC: Control transfer to illegal address 00011400

Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16638 Elapsed Time:00:00:00.0446227

Instructions per second:372859

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 :00000003

R1 :00000005

R2 :00000002

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) :00011400

R14 (lr) :00000000

R15 (pc) :00011400

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 :System

0x800000df

CodeView

Week2_Program2_PES2UG19CS075.o

.text

00001000:E3A00003 mov r0, #3

00001004:E3A01005 mov r1, #5

00001008:E1500001 cmp r0, r1 @unequal case

0000100C:0A000001 bEQ L1

00001010:E0412000 sub r2,r1,r0

00001014:EA000000 b end

L1:

00001018:E0812000 add r2,r1,r0

end:

0000101C:EF000011 swi 0x011

.end...

OutputView WatchView

Console stdin/stdout/stderr

Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program2_PES2UG19CS075.s

Execution starting ...

Location 113FC: Control transfer to illegal address 00011400

Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16639 Elapsed Time:00:00:00.0551132

Instructions per second:301905

Program Number: 3

Write an ALP to find the factorial of a number stored in R0. Store the value in R1 (without using LDR and STR instructions). Use only registers.

I. ARM Assembly Code for each program

```
[io] Week2_Program3_PES2UG19CS075.s
1  .text
2
3  mov r0, #3
4  mov r1, r0
5
6  fact:
7      sub r0, r0, #1
8      cmp r0, #1
9      bEQ end
10     mul r2, r1, r0
11     mov r1, r2
12     b fact
13
14 end:
15 .end
16
```

II. Output Screen Shot

The screenshot displays an ARM assembly simulator interface with three main panels: RegistersView, CodeView, and OutputView.

RegistersView: Shows the state of 16 registers (R0-R15) and CPSR. R0 is 00000001, R1 is 00000006, R2 is 00000006, and R15 (PC) is 00011400. CPSR shows Negative (N) as 0, Zero (Z) as 1, Carry (C) as 1, and Overflow (V) as 0.

CodeView: Displays the assembly code for 'Week2_Program3_PES2UG19CS075.o'. The code includes a 'fact' loop that calculates the factorial of the value in R0 (initially 3) and stores the result in R1. The code ends at address 00011400.

OutputView: Shows the console output, including the loading of the assembly file, execution starting, and a warning about control transfer to an illegal address (00011400). The execution ends with an instruction count of 16643 and an elapsed time of 00:00:00.0490046.

Program Number: 4a

Write an ALP to add two 32 bit numbers loaded from memory and store the result in memory.

I. ARM Assembly Code for each program

```
[18] Week2_Program4a_PES2UG19CS075.s
1  .data
2  A: .word 3
3  B: .word 4
4  C: .word 0
5
6  .text
7  ldr R0, =A
8  ldr R1, =B
9  ldr R2, =C
10 ldr R3, [R1]
11 ldr R4, [R0]
12 add R5, R3, R4
13 str R5, [R2]
14
15 .end
16
```

II. Output Screen Shot

The screenshot displays an ARM assembly debugger interface with three main panels: RegistersView, CodeView, and Console.

RegistersView: Shows the state of 16 ARM registers (R0-R15) and the CPSR register. R0-R4 contain 0x00000003, 0x00000004, 0x00000004, 0x00000003, and 0x00000007 respectively. R5-R15 are zero. The CPSR register shows flags: Negative (N): 0, Zero (Z): 0, Carry (C): 0, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, CPU Mode: System. The PC register (R15) is 0x000000df.

CodeView: Displays the assembly code for 'Week2_Program4a_PES2UG19CS075.o'. It shows the .data section with variables A (0x00000003), B (0x00000004), and C (0x00000000). The .text section contains instructions: ldr R0, =A; ldr R1, =B; ldr R2, =C; ldr R3, [R1]; ldr R4, [R0]; add R5, R3, R4; str R5, [R2]; and .end. Address ranges are provided for each instruction.

Console: Shows the execution log. It starts with 'Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program4a_PES2UG19CS075.s'. It then reports 'Execution starting ...'. Two warnings are shown: 'Location 113FC: Control transfer to illegal address 00011400' and 'Location 113FC: Control transfer to illegal address 00011400'. The execution ends with 'Execution ending, Instruction Count:16640 Elapsed Time:00:00:00.0480046' and 'Instructions per second:346633'.

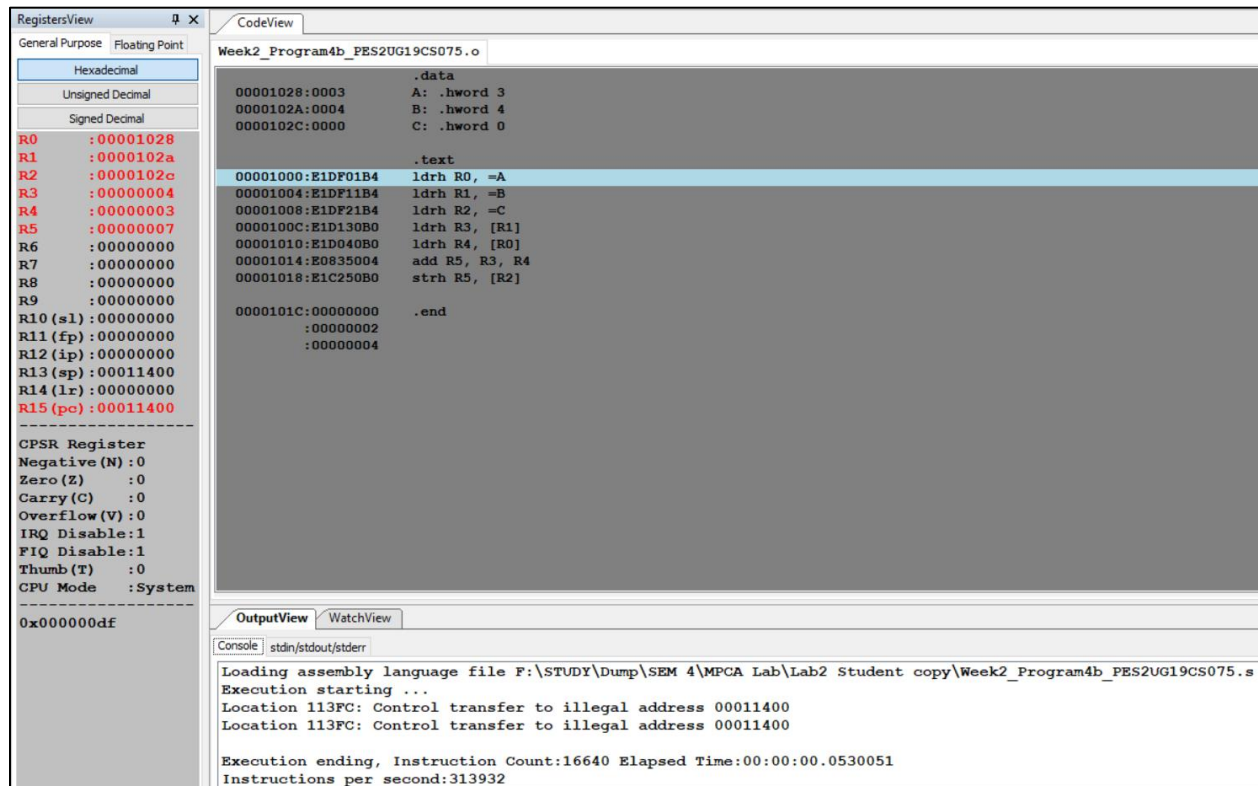
Program Number: 4b

Write an ALP to add two 16 bit numbers loaded from memory and store the result in memory.

I. ARM Assembly Code for each program

```
[10] Week2_Program4b_PES2UG19CS075.s
1  .data
2  A: .hword 3
3  B: .hword 4
4  C: .hword 0
5
6  .text
7  ldrh R0, =A
8  ldrh R1, =B
9  ldrh R2, =C
10 ldrh R3, [R1]
11 ldrh R4, [R0]
12 add R5, R3, R4
13 strh R5, [R2]
14
15 .end
16
```

II. Output Screen Shot



Program Number: 5a

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

I. ARM Assembly Code for each program

```
[a8] Week2_Program5a_PES2UG19CS075.s
1  .text
2  mov r0, #12
3  mov r1, #18
4  mov r2, R0
5  mov r3, r1
6
7  check:
8      cmp r2, r3
9      bEQ end
10     bLT less
11     b greater
12     less:
13         sub r3, r3, r2
14         b check
15     greater:
16         sub r2, r2, r3
17         b check
18
19     end:
20     .end
```

II. Output Screen Shot

The screenshot displays an ARM development environment with three main panels: RegistersView, CodeView, and OutputView.

RegistersView: Shows the state of 16 general-purpose registers (R0-R15) and the CPSR register. R0 is 0x0000000c, R1 is 0x00000012, R2 is 0x00000006, and R3 is 0x00000006. R10 (s1) through R15 (pc) are all 0x00000000. 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.

CodeView: Displays the assembly code for Week2_Program5a_PES2UG19CS075.o. The code is identical to the one shown in the previous block, starting with a .text section and ending with .end.

OutputView: Shows the execution output in the Console. The output indicates that the assembly language file was loaded successfully, execution started, and then encountered a control transfer to an illegal address (00011400) at location 113FC. The execution ended with an instruction count of 16645 and an elapsed time of 00:00:00.0510026 seconds.

Program Number: 5b

Write an ALP to find the GCD of given numbers (both numbers in memory) Store result in memory.

I. ARM Assembly Code for each program

```
Week2_Program5b_PES2UG19CS075.s
1  .data
2  A: .word 12
3  B: .word 18
4
5  .text
6  ldr r0, =A
7  ldr r1, =B
8  ldr r2, [r0]
9  ldr r3, [r1]
10
11  check:
12  cmp r2, r3
13  beq end
14  blt less
15  b greater
16  less:
17  sub r3, r3, r2
18  b check
19  greater:
20  sub r2, r2, r3
21  b check
22
23  end:
24  .end
25
```

II. Output Screen Shot

The screenshot displays an ARM assembly debugger interface. On the left, the 'RegistersView' pane shows the state of registers R0 through R15 and the CPSR register. R0-R15 are all zero, and CPSR flags are also zero. The top pane, 'CodeView', shows the assembly code for 'Week2_Program5b_PES2UG19CS075.o'. The code defines two words, A (12) and B (18), and implements a GCD algorithm using the Euclidean method. The bottom pane, 'OutputView', shows the console output, which includes the loading of the assembly file, the start of execution, a warning about control transfer to an illegal address (00011400), and the final execution statistics: 16645 instructions executed in 0.0510047 seconds, resulting in 326342 instructions per second.

```
RegistersView  X
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 (s1): 00000000
R11 (fp): 00000000
R12 (ip): 00000000
R13 (sp): 00011400
R14 (lr): 00000000
R15 (pc): 00011400
-----
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
Week2_Program5b_PES2UG19CS075.o
.data
00001038:0000000C A: .word 12
0000103C:00000012 B: .word 18
.text
00001000:E59F0028 ldr r0, =A
00001004:E59F1028 ldr r1, =B
00001008:E5902000 ldr r2, [r0]
0000100C:E5913000 ldr r3, [r1]
check:
00001010:E1520003 cmp r2, r3
00001014:0A000005 beq end
00001018:BA000000 blt less
0000101C:EA000001 b greater
less:
00001020:E0433002 sub r3, r3, r2
00001024:EAF0FF9F b check
greater:
00001028:E0422003 sub r2, r2, r3
0000102C:EAF0FF9F b check
end:
00001030:00000000 .end...
:00000004

OutputView  WatchView
Console | stdin/stdout/stderr
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab 2 Student copy\Week2_Program5b_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400
Execution ending, Instruction Count:16645 Elapsed Time:00:00:00.0510047
Instructions per second:326342
```

Program Number: 6a

Write an ALP to add an array of ten 32 bit numbers from memory.

I. ARM Assembly Code for each program

```
[10] Week2_Program6a_PES2UG19CS075.s
1  .data
2  A: .word 1,2,3,4,5,6,7,8,9,10
3  count: .word 10
4  sum: .word 0
5  .text
6  ldr r0, =A
7  ldr r1, =count
8  ldr r3, =sum
9  ldr r1, [r1]
10 ldr r3, [r3]
11 check:
12 ldr r2, [r0]
13 add r3, r3, r2
14 add r0, r0, #4
15 sub r1, r1, #1
16 cmp r1, #0
17 bEQ end
18 b check
19 end:
20 .end
21
```

II. Output Screen Shot

The screenshot displays an ARM assembly simulator interface with three main panels: RegistersView, CodeView, and OutputView.

RegistersView: Shows the state of various registers. R0 through R15 are listed with their values in hexadecimal. R15 (PC) is highlighted in red, showing the current instruction address 00011400. The CPSR Register is also shown with various flags like Negative (N), Zero (Z), Carry (C), Overflow (V), IRQ Disable, FIQ Disable, Thumb (T), and CPU Mode (System).

CodeView: Displays the assembly code for the program. The code is organized into sections: .data, .text, and .end. The .text section contains the main logic, including loading the array address (A) into R0, loading the count into R1, loading the sum into R3, and a loop that adds elements of the array to the sum until the count reaches zero.

OutputView: Shows the execution output. It includes the loading of the assembly language file, the start of execution, and a message indicating that execution is ending. The instruction count is 16702, and the elapsed time is 00:00:00.0510039. The instructions per second are 327465.

Program Number: 6b

Write an ALP to add array of ten 8 bit numbers taking data from memory location stored as byte data (use .byte to store the data instead of .word).

I. ARM Assembly Code for each program

```
[a0] Week2_Program6b_PES2UG19CS075.s
1  .data
2  A: .byte 1,2,3,4,5
3  count: .byte 5
4  sum: .byte 0
5  .text
6  ldr r0, =A
7  ldr r1, =count
8  ldr r3, =sum
9  ldrsb r1, [r1]
10 ldrsb r3, [r3]
11 check:
12     ldrsb r2, [r0]
13     add r3, r3, r2
14     add r0, r0, #1
15     sub r1, r1, #1
16     cmp r1, #0
17     bEQ end
18     b check
19 end:
20 .end
```

II. Output Screen Shot

The screenshot displays an ARM development environment with three main panels:

- RegistersView:** Shows the state of ARM registers. R0 is 00001041, R1 is 00000000, R2 is 00000005, R3 is 0000000f, R4 is 00000000, R5 is 00000000, R6 is 00000000, R7 is 00000000, R8 is 00000000, R9 is 00000000, R10 (s1) is 00000000, R11 (fp) is 00000000, R12 (ip) is 00000000, R13 (sp) is 00011400, R14 (lr) is 00000000, and R15 (pc) is 0000103c. 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.
- CodeView:** Displays the assembly code for Week2_Program6b_PES2UG19CS075.o. It shows the .data section with A: .byte 1,2,3,4,5, count: .byte 5, and sum: .byte 0. The .text section contains instructions for loading the array, calculating the sum, and branching.
- OutputView:** Shows the console output. It indicates that the assembly language file was loaded, execution started, and there was an access to unaligned memory location 0xF. The execution ended with an instruction count of 43 and an elapsed time of 00:00:00.0210025.

Program Number: 7

Write an ALP to multiply using barrel shifter.

$$35 * R0$$

I. ARM Assembly Code for each program

```
[10] Week2_Program7_PES2UG19CS075.s
1  .text
2  mov r0, #5
3  add r0, r0, r0, lsl #2
4  mov r1, r0
5  mov r0, r0, lsl #3
6  sub r0, r0, r1
7  .end
8
```

II. Output Screen Shot

The screenshot displays an ARM assembly debugger interface with three main panels:

- RegistersView:** Shows the state of ARM registers. R0 is 0x000000af, R1 is 0x00000019, and R15 (PC) is 0x00011400. The CPSR register shows flags like Negative (N), Zero (Z), Carry (C), Overflow (V), IRQ Disable, FIQ Disable, Thumb (T), and CPU Mode (System).
- CodeView:** Displays the assembly code for 'Week2_Program7_PES2UG19CS075.o'. The code includes instructions: `mov r0, #5`, `add r0, r0, r0, lsl #2`, `mov r1, r0`, `mov r0, r0, lsl #3`, `sub r0, r0, r1`, and `.end`.
- OutputView:** Shows the execution log. It reports: 'Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program7_PES2UG19CS075.s', 'Execution starting ...', 'Location 113FC: Control transfer to illegal address 00011400' (twice), and 'Execution ending, Instruction Count:16640 Elapsed Time:00:00:00.0565116 Instructions per second:294452'.

Program Number: 8

Write an ALP to evaluate the expression $(A+B) + (3*B)$, where A and B are memory location.

* Use LSL instruction for multiplication.

I. ARM Assembly Code for each program

```
[as] Week2_Program8_PES2UG19CS075.s
1  .data
2  A: .word 1
3  B: .word 2
4  C: .word 0
5  .text
6  ldr r0, =A
7  ldr r1, =B
8  ldr r2, =C
9  ldr r3, [r0]
10 ldr r4, [r1]
11 add r5, r3, r4
12 mov r6, r4, lsl #2
13 add r6, r4, r6
14 add r7, r6, r5
15 str r7, [r2]
16 .end
17
```

II. Output Screen Shot

The screenshot displays an ARM assembly debugger interface with three main panels: RegistersView, CodeView, and OutputView.

RegistersView: Shows the state of 16 registers (R0-R15) and the CPSR register. R0-R15 are in hexadecimal format. R0 is 00001034, R1 is 00001038, R2 is 0000103c, R3 is 00000001, R4 is 00000002, R5 is 00000003, R6 is 0000000a, R7 is 0000000d, R8-R15 are 00000000. CPSR Register shows Negative (N): 0, Zero (Z): 0, Carry (C): 0, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, CPU Mode: System.

CodeView: Shows the assembly code for Week2_Program8_PES2UG19CS075.o. The code is as follows:

```
.data
00001034:00000001 A: .word 1
00001038:00000002 B: .word 2
0000103c:00000000 C: .word 0
.text
00001000:E59F0020 ldr r0, =A
00001004:E59F1020 ldr r1, =B
00001008:E59F2020 ldr r2, =C
0000100c:E5903000 ldr r3, [r0]
00001010:E5914000 ldr r4, [r1]
00001014:E0835004 add r5, r3, r4
00001018:E1A06104 mov r6, r4, lsl #2
0000101c:E0846006 add r6, r4, r6
00001020:E0867005 add r7, r6, r5
00001024:E5827000 str r7, [r2]
00001028:00000000 .end
:00000004
:00000008
```

OutputView: Shows the console output of the program execution. The output is as follows:

```
Loading assembly language file F:\STUDY\Dump\SEM 4\MPCA Lab\Lab2 Student copy\Week2_Program8_PES2UG19CS075.s
Execution starting ...
Location 113FC: Control transfer to illegal address 00011400
Location 113FC: Control transfer to illegal address 00011400

Execution ending, Instruction Count:16640 Elapsed Time:00:00:00.0550048
Instructions per second:302519
```

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: *Atul Anurag*

Name: Atul Anurag

SRN: PES2UG19CS075

Section: B

Date: 03-02-2021