Lab Task (Psychomotor)

Activity 01:

Build and Run the following code:

```
    EXPORT SystemInit
    EXPORT __main


    AREA MDATA1, DATA, READONLY
DATA1 DCD 7

    AREA MDATA2, DATA, READWRITE
DATA2 DCD 7

    AREA PROGRAM, CODE, READONLY

SystemInit FUNCTION

    BX      LR                  ;Return from function

    ENDFUNC

__main FUNCTION

    LDR R0,=DATA1
    LDR R1,[R0]

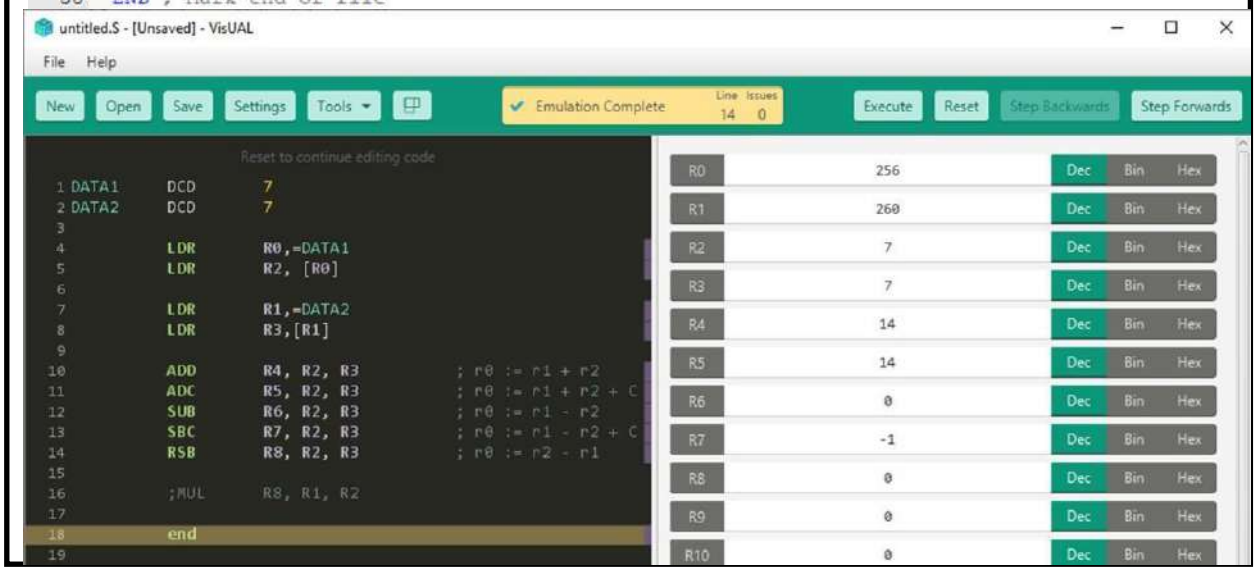
    LDR R3,=DATA2
    LDR R2,[R3]

    ADD R4, R1, R2      ; r0 := r1 + r2
    ADC R5, R1, R2      ; r0 := r1 + r2 + C
    SUB R6, R1, R2      ; r0 := r1 - r2
    SBC R7, R1, R2      ; r0 := r1 - r2 + C - 1
    RSB R8, R1, R2      ; r0 := r2 - r1

;   MUL R8, R1, R2
;   RSC R0, R1, R2      ; r0 := r2 - r1 + C - 1

ENDFUNC

END ; Mark end of file
```



Activity 02:

Calculate the following expression by initializing four variables.

[(a*b) + (c*d)]

Lab Task (Cognitive)

Question 1:

Consider the following ARM program fragment:

```
LDR   R0, =0x7FFFFFFF
MOVS  R1, #1
ADDS  R2, R0, R1
SUBS  R3, R1, R1
ANDS  R4, R0, R1
ORRS  R5, R0, R1
EORS  R6, R0, R1
```

a) Determine the value of R2 after ADDS and the state of the N, Z, C and V flags immediately after it. **[4 Marks]**

b) Determine the value of R3 after SUBS and the state of the Z and C flags immediately after it. **[3 Marks]**

c) Determine the values of R4, R5 and R6 in hexadecimal. **[3 Marks]**

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature: _____ **Date:** _____

LABORATORY SKILLS ASSESSMENT (Cognitive)

Obtained Marks	**/10**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Psychomotor Activity 1: Experimental Setup

S. No	Criteria	Performance	Description	Marks
1	**Keil Assembly Setup (4)**	Level 4	Correctly prepares the required Keil assembly setup for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required Keil assembly setup with minor errors or assistance.	3
		Level 2	Partially prepares the required Keil assembly setup and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required Keil assembly setup and requires substantial assistance.	1
2	**Arithmetic Program & Register Setup (4)**	Level 4	Correctly prepares the required arithmetic program and register setup for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required arithmetic program and register setup with minor errors or assistance.	3
		Level 2	Partially prepares the required arithmetic program and register setup and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required arithmetic program and register setup and requires substantial assistance.	1
3	**Debug & Flag-Monitoring Parameters (2)**	Level 2	Correctly sets the required debug and flag-monitoring parameters for the experiment.	2
		Level 1	Sets the basic debug and flag-monitoring parameters with minor errors or assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Instructor's Signature: _____ Date: _____

Psychomotor Activity 2: Implementation + Results

S. No	Criteria	Performance	Description	Marks
1	**Arithmetic & Logical Instructions (4)**	Level 4	Correctly implements the assigned arithmetic and logical instructions independently while following appropriate laboratory practices.	4
		Level 3	Implements the assigned arithmetic and logical instructions with minor errors or assistance and generally follows proper laboratory practices.	3
		Level 2	Partially implements the assigned arithmetic and logical instructions and requires considerable assistance or repeated guidance.	2
		Level 1	Implementation of the assigned arithmetic and logical instructions is largely incomplete, requires substantial assistance, or shows poor adherence to proper laboratory and safety practices.	1
2	**Execution & Debugging (2)**	Level 2	Correctly performs the required execution and debugging and verifies the expected operation.	2
		Level 1	Performs the basic execution and debugging with minor errors or assistance.	1
3	**Register & Flag (APSR) Analysis (4)**	Level 4	Obtains correct register and flag (APSR) analysis and appropriately verifies and interprets the required experimental results.	4
		Level 3	Obtains mostly correct register and flag (APSR) analysis with minor errors in verification or interpretation.	3
		Level 2	Obtains partially correct register and flag (APSR) analysis with noticeable errors or omissions.	2
		Level 1	Register and flag (APSR) analysis is largely incorrect or incomplete and requires substantial assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Affective)

Affective Activity: Lab Report + Viva Voce

S. No	Criteria	Performance	Description	Marks
1	**Lab Report Completeness & Organization (4)**	Level 4	Report is complete, well-organized, properly presented, and demonstrates responsible completion of the assigned laboratory work.	4
		Level 3	Report is complete with minor omissions or organizational issues and demonstrates satisfactory responsibility for the assigned work.	3
		Level 2	Report is partially complete with several omissions and demonstrates limited responsibility for the assigned work.	2
		Level 1	Report has major omissions or poor organization and demonstrates minimal responsibility for the assigned work.	1
2	**Register/Flag Output & Conclusion (2)**	Level 2	The register/flag output and conclusion are correctly presented and discussed, demonstrating ownership of the assigned work.	2
		Level 1	The register/flag output and conclusion are presented with minor errors or omissions.	1
	Viva Voce (4)	Level 4	Clearly explains the assigned experiment, arithmetic and logical instructions, status flags, program explanation, working, and results, demonstrating individual responsibility, participation, and effective communication.	4
		Level 3	Explains most of the assigned experiment and results correctly, with minor difficulty, and demonstrates good individual participation.	3
		Level 2	Explains basic aspects of the assigned experiment with some prompting and demonstrates basic participation.	2
		Level 1	Demonstrates limited understanding or involvement and requires substantial prompting to explain the assigned work.	1
Total Obtained				**/10**

The student has not submitted the report and has not appeared in the Viva Voce.

Criteria	Description	Marks
Lab Report and Results Not Submitted	Student has not submitted the report and results file.	**0**
Absent in Viva Voce	Student is absent from the viva voce.	**0**

Instructor's Signature: _____ Date: _____

Experiment No 5

ARM assembly programming using control flow Instructions

Objectives:

- To learn the basic of various flow control Instructions in ARM Architecture
- Use these flow control instructions to perform conditional and repetitive tasks
- Operation of user defined subroutines in ARM Assembly

Flow control, or branching, instructions for jumps, loops and conditional jumps allow us to control the execution of program statements and to build high-level programming structures such as if-else statements, case statements, and loops in assembler programs.

CMP Instruction:

The CMP (Compare) instruction compares two operands and effects the flags only. CMN(Compare Negative) is also a similar instruction but it is used to compare with Negative numbers. The syntax of CMP instruction is as follows:

CMP{cond} Rn, Operand2

where:

cond

is an optional condition code.

Rn

is the ARM register holding the first operand

Operand2

is a flexible second operand

The instruction compares the value in a register with 'Operand2'. They update the condition flags on the result, but do not place the result in any register. The CMP instruction subtracts the value of 'Operand2' from the value in 'Rn'. This is the same as a SUBS instruction, except that the result is discarded. In ARM Assembly, CMP instruction is used before conditional execution of instructions or conditional branch. The flags (N, Z, C, V) are only affected by this instruction

TEQ Instruction:

The TEQ (Test Equivalence) instruction compares two operands by doing the 'AND' operation between two 32-Bit registers and effects the flags only. The syntax of TEQ instruction is as follows:

TEQ{cond} Rn, Operand2

where:

cond
is an optional condition code.

Rn
is the ARM register holding the first operand

Operand2
is a flexible second operand

The instruction compares the value in a register with 'Operand2'. They update the condition flags on the result, but do not place the result in any register. The TEQ instruction performs 'AND' operation between 'Rn' and 'Operand2'. In ARM Assembly, TEQ instruction is used before conditional execution of instructions or conditional branch. The flags (N, Z) are only affected by this instruction.

Conditions in ARM Assembly:

ARM, like many other architectures, implements conditional execution using a set of flags whichstore state information about a previous operation. The simplest way to set the condition flags is to use a comparison operation, such as CMP. This mechanism is common to many processor architectures, and the semantics (if not the details) of CMP will likely be familiar. In addition, we have already seen that many instructions (such as SUB in the example) can be modified to update the condition flags based on the result by adding an 'S' suffix.

Conditional execution can be done through:

1. Conditional Instructions.
2. Branch Instructions.

Conditional Instructions:

All the ARM Assembly instructions can be executed by adding a suffix to the instruction.

For Example:

```
CMP     R2, #0
ADDEQ   R2, R2, #50     ; Add if TRUE
MULEQ   R5, R5, #8      ; Multiply if TRUE
LSRNE   R8, R8, #4      ; Shift Right if TRUE
SUBLE   R3, R3, #5      ; Subtract if TRUE
```

These instructions will only execute if the condition is satisfied.

Figure 1: Suffix for Conditional Instruction

Suffix	Description	Flags tested
EQ	Equal (Unsigned & Signed)	Z=1
NE	Not Equal (Unsigned & Signed)	Z=0
MI	MInus (Negative)	N=1
PL	PLus (Positive or Zero)	N=0
VS	oVerflow Set	V=1
VC	oVerflow Clear	V=0
HI	Unsigned HIgher	C=1 & Z=0
LS	Unsigned Lower or Same	C=0 or Z=1
GE	Signed Greater or Equal	N=V
LT	Signed Less Than	N!=V
GT	Signed Greater Than	Z=0 & N=V
LE	Signed Less than or Equal	Z=1 or N!=V
AL	ALways	

EXAMPLE:

```
1       EXPORT SystemInit
2       EXPORT __main
3
4           AREA  myData, DATA, READWRITE
5   Array2 DCD 0,0,0,0,0
6
7           AREA     MDATA1, DATA, READONLY
8   Array1 DCD 4,-5,9,-10,8
9
10          AREA    MYCODE, CODE, READONLY
11
12  SystemInit FUNCTION
13      BX       LR
14
15      ENDFUNC
16
17  __main FUNCTION
18
19      LDR      R0,=Array1
20      LDR      R1,=Array2
21      mov      R2, #0
22
23      LDR      R4,[R0],#4
24      CMP      R4, #0
25      SUBLT    R4, R2, R4
26      STR      R4,[R1],#4
27
28      LDR      R4,[R0],#4
29      CMP      R4, #0
30      SUBLT    R4, R2, R4
31      STR      R4,[R1],#4
32
33      LDR      R4,[R0],#4
34      CMP      R4, #0
35      SUBLT    R4, R2, R4
36      STR      R4,[R1],#4
37
38      LDR      R4,[R0],#4
39      CMP      R4, #0
40      SUBLT    R4, R2, R4
41      STR      R4,[R1],#4
42
43      LDR      R4,[R0],#4
44      CMP      R4, #0
45      SUBLT    R4, R2, R4
46      STR      R4,[R1],#4
47
48      ;===================================
49
50
51      ENDFUNC
52      END
53
```