-

Table of Contents

Note : Remove “Table of Content” before including in CP Book

18CSL48 : MICROCONTROLLER AND EMBEDDED SYSTEMS LAB

A. LABORATORY INFORMATION

1. Lab Overview

Degree:	BE	*Program:*	CS
Year / Semester :	2 / 4	*Academic Year:*	2018-19
CourseTitle:	Microcontroller and Embedded Systems lab	*Course Code:*	18CSL48
Credit / L-T-P:	2 / 1-0-2	*SEE Duration:*	180 Minutes
Total Contact Hours:	40 Hrs	*SEE Marks:*	60Marks
CIA Marks:	40	*Assignment*	1 / Module
Course Plan Author:	Prof. VINAY KUMAR B C	*Sign*	Dt :
Checked By:		*Sign*	Dt :

2. Lab Content

EXPT	**Title of the Experiments**	**Lab Hours**	**Concept**	**Blooms Level**
1	Write a program to multiply two 16 bit binary numbers.	03	**ALP**	L4 Analyze
2	Write a program to find the sum of first 10 integer numbers.	03	**ALP**	L4 Analyze
3	Write a program to find factorial of a number.	03	**ALP**	L4 Analyze
4	Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM	03	**ALP**	L4 Analyze
5	Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM	03	Interrupt Functions	L4 Analyze
6	Write a program to find the largest/smallest number in an array of 32 numbers	03	ARM ALP &C	L4 Analyze
7	Write a program to arrange a series of 32 bit numbers in ascending/descending order			
8	Write a program to count the number of ones and zeros in two consecutive memory locations			
	To write and simulate C Programs for ARM microprocessor using KEIL (Demonstrate with the help of a suitable program)	03	ARM ALP &C	L4 Analyze
9	Display “Hello World” message using Internal UART	03	I/O Interfacing	L4 Analyze
10	Interface and Control a DC Motor Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.	03	I/O Interfacing	L4 Analyze
11	Determine Digital output for a given Analog input using Internal ADC of ARM controller. Interface a DAC and generate Triangular and Square waveforms.	03	/O Interfacing	L4 Analyze
12	Interface a 4x4 keyboard and display the key code on an LCD	03	/O Interfacing	L4 Analyze
13	Interface a 4x4 keyboard and display the key code on an LCD On/ Off.	03	ARM Interface	L4 Analyze
14	Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between	03	ARM Interface	L4 Analyze

3. Lab Material

Unit	Details	Available
1	Text books	
		In Lib

2	Reference books	
		In dept
3	Others (Web, Video, Simulation, Notes etc.)	
		Not Available

4. Lab Prerequisites:

-	-	Base Course:		-	-
SNo	Course Code	Course Name	Topic / Description	Sem	Remarks
1	18CSL48	Microcontroller and Embedded System Lab		2	
				2	

Note: If prerequisites are not taught earlier, GAP in curriculum needs to be addressed. Include in Remarks and implement in B.5.

5. General Instructions

SNo	Instructions	Remarks
1	Observation book and Lab record are compulsory.	
2	Students should report to the concerned lab as per the time table.	
3	After completion of the program, certification of the concerned staff in-charge in the observation book is necessary.	
4	Student should bring a notebook of 100 pages and should enter the readings /observations into the notebook while performing the experiment.	
5	The record of observations along with the detailed experimental procedure of the experiment in the Immediate last session should be submitted and certified staff member in-charge.	
6	Should attempt all problems / assignments given in the list session wise.	
7	It is responsibility to create a separate directory to store all the programs, so that nobody else can read or copy.	
8	When the experiment is completed, should disconnect the setup made by them, and should return all the components/instruments taken for the purpose.	
9	Any damage of the equipment or burn-out components will be viewed seriously either by putting penalty or by dismissing the total group of students from the lab for the semester/year	
10	Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, program code along with comments and output for various inputs given	

6. Lab Specific Instructions

SNo	Specific Instructions/Software Programs 8086	Remarks
1	Open DOS editor to create file	
2	Use KEIL tool to Assemble , debug and execute file	
	Specific Instructions/Software Programs ARM	

1	Use KEIL tool to Assemble , debug and execute file	
	Specific Instructions/Hardware Programs 8086	
1	Open DOS editor to create file	
2	Use KEIL tool to Assemble , debug and execute file	
3	Do connections as per interface diagram	
4	Test results	
	Specific Instructions/Hardware Programs ARM	
1	Use KEIL tool to Assemble , debug and execute file	
2	Do connections as per interface diagram	
3	Test results	

B. OBE PARAMETERS

1. Lab / Course Outcomes

#	COs	Teach. Hours	Concept	Instr Method	Assessment Method	Blooms' Level
1	Develop and test program using ARM7TDMI/LPC2148	24	ALP	Instructions &Demonstration	Slip Test	L2
2	Conduct the following experiments on an ARM7TDMI/LPC2148 evaluation board using evaluation version of Embedded 'C' & Keil Uvision-4 tool/compiler	16	INTERFACING	Instructions &Demonstration	Slip Test	L3
-	**Total**	**39**	-	-	-	-

Note: Identify a max of 2 Concepts per unit. Write 1 CO per concept.

2. Lab Applications

SNo	Application Area	CO	Level
1	Assembly language programming is required to develop system programs	CO1	L2
2	Interrupt Functions is required to formulate system program solutions	CO1	L2
3	ARM AL and 'C' programming is required to develop embedded systems.	CO2	L3
4	ARM programming for interfacing external devices is used design and develop embedded systems.	CO2	L3

Note: Write 1 or 2 applications per CO.

3. Articulation Matrix

(CO – PO MAPPING)

-	Course Outcomes	Program Outcomes												
#	COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	Level
18CSl48.1	Develop and test program using ARM7TDMI/LPC2148	3	2	3		3								2
18CSl48.2	Conduct the following experiments on an ARM7TDMI/LPC2148 evaluation board using evaluation version of Embedded 'C' & Keil Uvision-4 tool/compiler.	3	2	3		3						2	2	3

Note: Mention the mapping strength as 1, 2, or 3

4. Mapping Justification

Mapping		Mapping Level	Justification
CO	**PO**	**-**	**-**
CO1	PO1	3	Knowledge of assembly language programming is required to develop system programs
CO1	PO2	2	Knowledge of assembly language programming is useful in analyzing system programs
CO1	PO3	3	Assembly language programs used to design and develop system programs
CO1	PO5	3	Assembler tool used to learn Assembly language programming
CO2	PO1	3	Knowledge of Interrupt Functions is required to develop system programs
CO2	PO2	2	Interrupt Functions is required to formulate system program solutions
CO2	PO3	3	Interrupt Functions are used to design and develop system programs
CO2	PO5	3	Interrupt Functions help in development of system program projects
CO2	PO11	2	Learning in the context of technology changes
CO2	PO12	3	ARM programming for interfacing external devices is used design and develop embedded systems.

Note: Write justification for each CO-PO mapping.

5. Curricular Gap and Content

SNo	Gap Topic	Actions Planned	Schedule Planned	Resources Person	PO Mapping
1					
2					
3					
4					
5					

Note: Write Gap topics from A.4 and add others also.

6. Content Beyond Syllabus

SNo	Gap Topic	Actions Planned	Schedule Planned	Resources Person	PO Mapping
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

Note: Anything not covered above is included here.

C. COURSE ASSESSMENT

1. Course Coverage

Unit	Title	Teaching Hours	No. of question in Exam							CO	Levels
			CIA-1	CIA-2	CIA-3	Asg-1	Asg-2	Asg-3	SEE		
1	ALP for 16 Bit addition	03	1	-	-	-	-	-	1	CO1	
2	ALP for first addition of ten numbers	03	1	-	-	-	-	-	1	CO1	
3	ALP for factorial number	03	1	-	-	-	-	-	1	CO1	
4	ALP for add an array of 16 bit numbers	03	1		-	-	-	-	1	CO1	
5	ALP to square of a number (1 to 10) using look-up table. Write a program to count the number of ones and zeros in two consecutive memory locations	03	-	1	-	-	-	-	1	CO1	
6	largest/smallest number in an array of 32 numbers arrange a series of 32 bit numbers in ascending/descending order	03	-	1	-	-	-	-	1	CO1	
7	ARM C programs	03	-	1	-	-	-	-	1	CO2	
8	Display "Hello World" message using Internal UART.	03	-	1		-	-	-	1	CO2	
9	Determine Digital output for a given Analog input using Internal ADC of ARM controller	03	-	-	1	-	-	-	1	CO2	
10	interface stepper motor Interface and Control a DC Motor	03	-	-	1	-	-	-	1	CO2	
11	ALP to interface DAC Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between	03	-	-	1	-	-	-	1	CO2	
12	interface LCD on/off	03	-	-	1	-	-	-	1	CO2	
13	Interface a 4x4 keyboard and display the key code on an LCD	03	-	-	1	-	-	-	1	CO2	
-	**Total**	**39**	**4**	**4**	**5**				**13**	**-**	

Note: Write CO based on the theory course.

2. Continuous Internal Assessment (CIA)

Evaluation	Weightage in Marks	CO	Levels
CIA Exam – 1	25	CO1	L2
CIA Exam – 2	25	CO1	L2
CIA Exam – 3	25	CO1,CO2	L3
record	15		
Final CIA Marks	**40**	**-**	**-**

-

SNo	Description	Marks
1	Observation and Weekly Laboratory Activities	05 Marks
2	Record Writing	10 Marks for each Expt
3	Internal Exam Assessment	25 Marks

IS

Prepared by Checked by Approved

4	Internal Assessment	40 Marks
5	SEE	60 Marks
-	**Total**	**100 Marks**

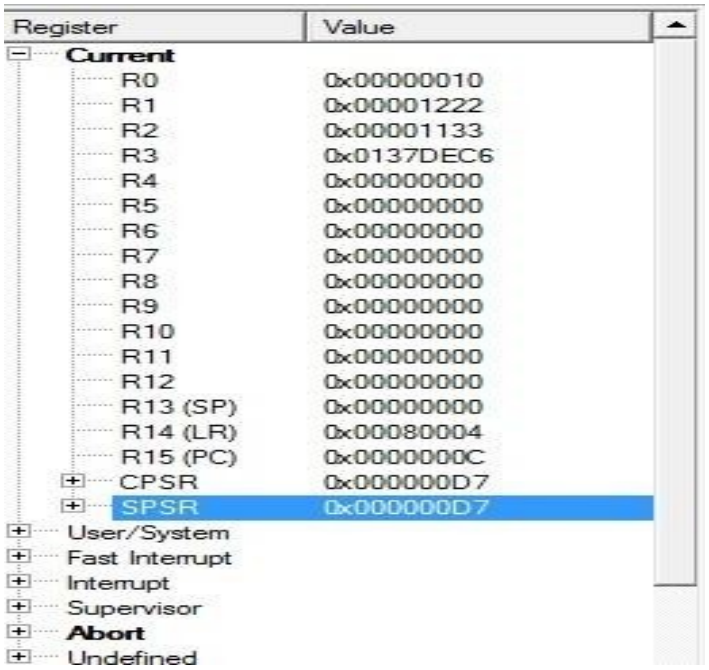
D. EXPERIMENTS

Experiment 01 :

-	**Experiment No.:**	1	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	ALP for 16 Bit addition						
2	Course Outcomes	Able to develop ARM Assembly language program for addition						
3	Aim	Write a program to multiply two 16 bit binary numbers.						
4	Material / Equipment Required	1. Designing tool software KEIL						
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code1	. AREA Multiply, CODE, READONLY
		ENTRY
		LDR R0, =NUM ; load address of multiplicand
		LDRH R1, [R0] ; load First number
		
		LDRH R2, [R0,#2] ; load Second number
		MUL R3, R1, R2 ; R3 = R1 x R2
		STOP B STOP ; all done
		NUM DCW 0X1222,0X1133 ; Declaration of no's to be multiply
		
		END						

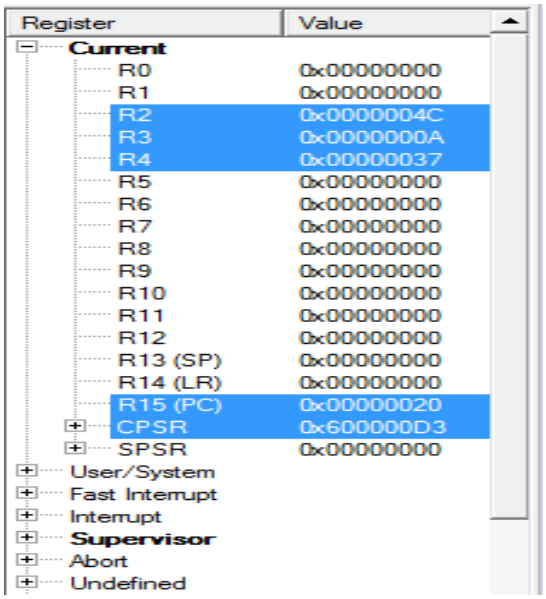
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output							
9	Sample Calculations							
10	Graphs, Outputs							

		
11	Results &Analysis	
12	Application Areas	Assembly language programming is required to develop system programs
13	Remarks	
14	Faculty Signature with Date	

Experiment 02 :

-	**Experiment No.:**	2	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	first addition of ten numbers						
2	Course Outcomes	Able to develop ARM Assembly language programfor addition first ten numbers						
3	Aim	**1.** Write a program to find the sum of first 10 integer numbers.						
4	Material / Equipment Required	designing tool software keil						
5	Theory, Formula, Principle, Concept	Able to develop ARM Assembly language program						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	AREA ADD1TO10, CODE, READONLY ENTRY
		MOV R1,#10 ;length of array
		LDR R2,=ARRAY ;Load the starting address of the array
		MOV R4,#0 ;Initial sum
		NEXT LDR R3,[R2],#4 ;Load first integer of the
		array in R3 ADD R4,R4,R3 ;R4=sum of integers
		SUBS R1,R1,#1
		BNE NEXT						

		;repeat until R1=0 STOP B STOP ARRAY DCD 1,2,3,4, 5,6,7,8, 9,10 END
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results &Analysis	
12	Application Areas	Assembly language programming is required to develop system programs
13	Remarks	
14	Faculty Signature with Date	

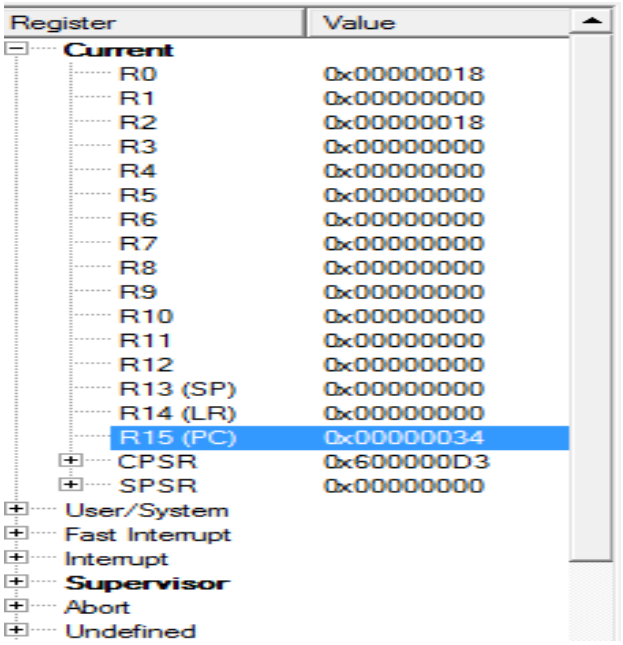
Experiment 03 :

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	

1	Title	Factorial of a number
2	Course Outcomes	Able to develop ARM Assembly language program for FACTORIAL
3	Aim	**2.** Write a program to find factorial of a number.
4	Material / Equipment Required	1. Designing tool software KEIL
5	Theory, Formula, Principle, Concept	Able to develop ARM Assembly language program

6	Procedure, Program, Activity, Algorithm, Pseudo Code	AREA Factorial, CODE, READONLY
		ENTRY
		MOV R0,#4 ; load the number in R0
		CMP R0,#0 ; check if the number is 0
		BEQ ANS ;if number is 0, go to label ANS
		CMP R0,#1 ; check if the number is 1
		BEQ ANS ;if number is 1, go to label ANS
		MOV R1,R0 ; Copy the number in R1
		UP SUBS R1,R1,#1 ; decrement the value
		in R1 till 0 BEQ STOP ; if yes store
		factorial value
		MUL R2,R1,R0 ; if not fact= R0 x R1
		MOV R0,R2 ; move fact value
		B UP
		;repeat until R1 is 0 ANS
		MOV R0,#1
		STOP B STOP ; Stop

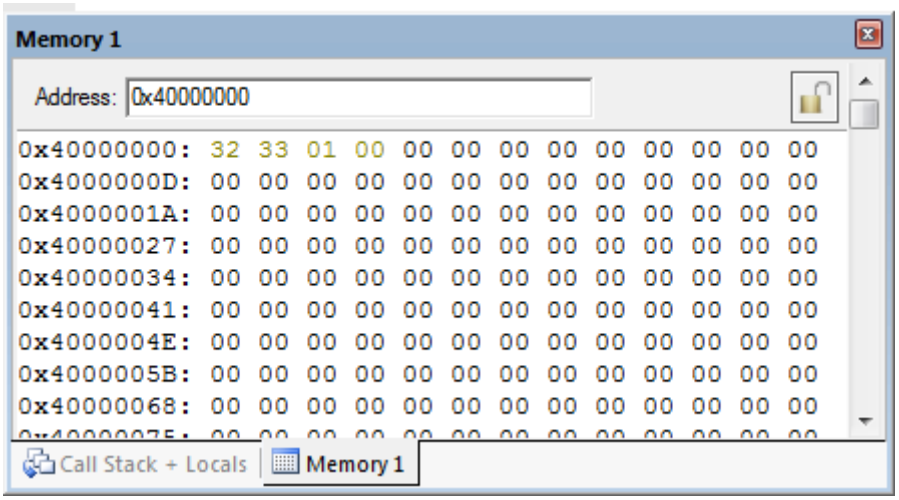
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	

		
11	Results &Analysis	
12	Application Areas	Assembly language programming is required to develop system programs
13	Remarks	
14	Faculty Signature with Date	

Experiment 04 : .....

-	**Experiment No.:**	4	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	add an array of 16 bit numbers						
2	Course Outcomes	Able to develop ARM Assembly language program						
3	Aim	Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM						
4	Material / Equipment Required	1. Designing tool software KEIL						
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	AREA ADDITION,CODE,READONLY ENTRY
		MOV R5,#6 ;length of array
		MOV R0,#0 ;initial sum
		LDR R1,=VALUE1 ;starting address of the array LOOP LDRH R2,[R1],#2 ;R2=first element of array						

		ADD R0,R0,R2 ;add first element with initial sum SUBS R5,R5,#1 BNE LOOP ;repeat addition until r5=0 LDR R4,=RESULT STR R0,[R4] ;store the result in memory STOP B STOP VALUE1 DCW 0X1111,0X2222,0X3333,0X4444,0X3333 ,0X5555 AREA DATA2,DATA,READWRITE RESULT DCD 0X0 END
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results &Analysis	
12	Application Areas	Assembly language programming is required to develop system programs
13	Remarks	
14	Faculty Signature with Date	

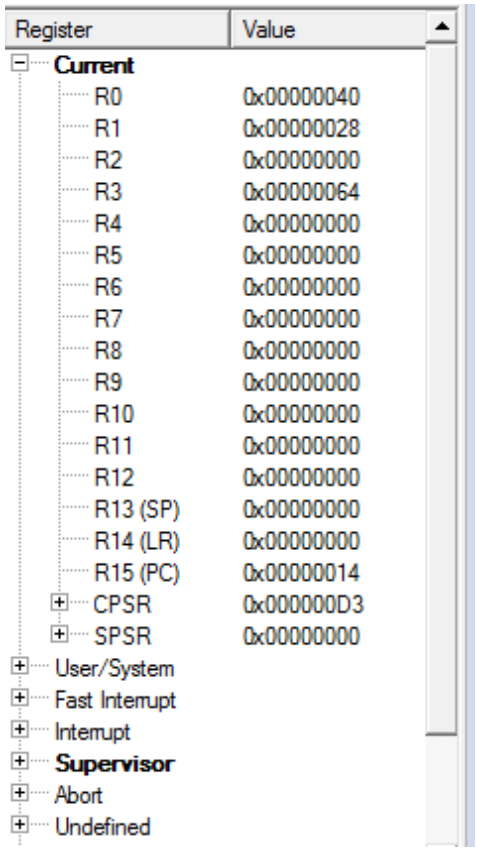
Experiment 05 : .....

-	**Experiment No.:**	5	**Marks**		**Date**		**Date**	

					Planned		Conducted	
1	Title	square of a number (1 to 10) using look-up table.						
2	Course Outcomes	Able to develop ARM Assembly language program using LOOKUP TABLE						
3	Aim	Write a program to find the square of a number (1 to 10) using look-up table.						
4	Material / Equipment Required	1. Designing tool software KEIL						
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	AREA square, CODE, READONLY ENTRY
		MOV R1,#0X3 ; load the number to be squared
		LDR R0,=LOOKUP ; load the starting address of the lookup table
		MOV R1,R1,LSL#0X2 ; offset of value to be squared
		ADD R0,R0,R1 ; points to mem where square of the given no is so
		LDR R3,[R0] ; load the squared value from look-up table
		STOP B STOP
		LOOKUP DCD
		0X0,0X1,0x4,0x9,0x10,0x19,0x24,0x31,0x40,0x51,0x64
		; look-up table
		E
		N
		D						

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output							

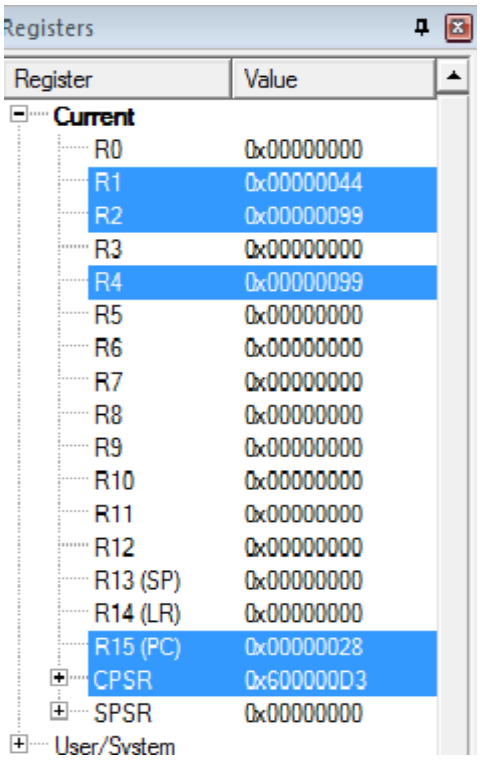
9	Sample Calculations	
10	Graphs, Outputs	
11	Results &Analysis	
13	Remarks	Interrupt Functions is required to formulate system program solutions
14	Faculty Signature with Date	

Experiment 06 : .....

-	**Experiment No.:**	6	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	largest/smallest number in an array of 32 numbers..						
2	Course Outcomes	Able to simulate ARM assembly language for data transfer, arithmetic and logical operations						
3	Aim	**3.** Write a program to find the largest/smallest number in an array of 32 numbers.						
4	Material / Equipment Required	1. Designing tool software KEIL						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo	AREALARGE,CODE,READONLY ENTRY						

	Code	MOV R5,#5 ;R5 = length of array - 1
		LDR R1,=ARRAY ;load starting
		addressing of array LDR R2,[R1],#4
		;load 1st element of
		array
		LOOP LDR R4,[R1],#4 ;load next element
		of array CMP R2,R4 ;compare
		1st and 2nd element BHI NEXT
		MOV R2,R4 ;R2=largest value
		NEXT SUBS R5,R5,#1 ;decrement the counter after every
		comparison BNE LOOP ; repeat until R5=0
		STOP B STOP
		
		ARRAY DCD
		
		0X23,0X45,0X65,0X76
		
		,0X12,0X99
		
		END

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
	Observation Table, Look-up Table, Output	
	Sample Calculations	
10	Graphs, Outputs	

		
11	Results &Analysis	
12	Application Areas	ARM AL and 'C' programming is required to develop embedded systems.
13	Remarks	
14	Faculty Signature with Date	

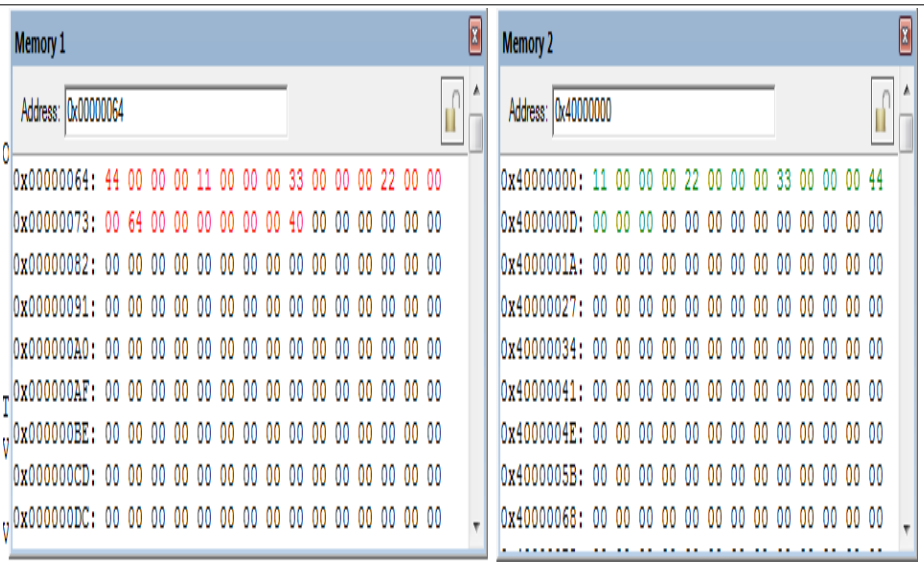
Experiment 07 : .....

-	**Experiment No.:**	7	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Arrange a series of 32 bit numbers in ascending/descending order.						
2	Course Outcomes	Able to simulate ARM assembly language and 'C'programs for data transfer, arithmetic and logical operations						
3	Aim	Write a program to arrange a series of 32 bit numbers in ascending/descending order.						
4	Material / Equipment Required	1. Designing tool software KEIL.						
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	AREA Ascending,
		CODE,
		READONLY
		ENTRY
		MOV R8,#4 ;Length of the array
		LDR R2,=SVALUE ;Starting address of the
		source array LDR R3,=DVALUE ;Starting						

		address of the destination array
		LOOP0 LDR R1,[R2],#4 ;Loop0 copies all the elements of source ary to dest ary STR R1,[R3],#4
		MOV R7,#3
		;R7=Number of pass NXTPAS
		MOV R5,R7
		;R5=Number of comparisons
		LDR R1,=DVALUE ;Loads the starting address of dest array in R1 NXTCMP LDR R2,[R1],#4
		LDR R3,[R1]
		CMP R2,R3 ;Compares first and second element of the array
		BLT NOSWP ;If first element is smaller, no swapping
		STR R2,[R1],#-4 ;Swaps the elements of the array STR R3,[R1]
		ADD R1,R1,#4
		NOSWP SUBS R5,R5,#1 ;Decrement comparison counter by 1 till 0 BNE NXTCMP
		SUBS R7,R7,#1 ;Decrement pass counter by 1 till 0 BNE NXTPAS
		STOP B STOP
		SVALUE DCD 0X44,0X11,0X33,0X22
		AREA DATA1,DATA,READWRITE
		DVALUE DCD 0X00
		END

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	

		
11	Results &Analysis	
12	Application Areas	ARM AL and 'C' programming is required to develop embedded systems.
13	Remarks	
14	Faculty Signature with Date	

PART B

Experiment 09: .....

-	**Experiment No.:**	8a	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	**1.** Display "Hello World" message using Internal UART.						
2	Course Outcomes	Able to Design and develop ARM programs to interface with external I/O devices						
3	Aim	**2.** Display "Hello World" message using Internal UART.						
4	Material / Equipment Required							
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code							

```
#include <lpc214x.h>
void
uart_interrupt(void)
irq ; unsigned char
temp , temp1 =
0x00 ; unsigned
char rx_flag = 0 ,
tx_flag = 0 ;

int main(void)
{
PINSEL0=0X0000005;                  //select TXD0 and RXD0 lines

U0LCR = 0X00000083;                 //enable baud rate
divisor loading and U0DLM = 0X00;          //select the
data format
U0DLL = 0x13;                              //select
baud rate 9600 bps U0LCR = 0X00000003;
U0IER = 0X03;                              //select Transmit and Recieve interrupt

   VICVectAddr0 = (unsigned long)uart_interrupt;        //UART 0
   INTERRUPT VICVectCntl0 = 0x20|6;                     // Assign the
   VIC channel uart-0 to interrupt priority 0
VICIntEnable = 0x00000040;      // Enable the uart-0 interrupt
   while(1)
   {
   while(rx_flag == 0x00);              //wait for
receive flag to set rx_flag = 0x00;  //clear the
flag

U0THR = temp1 ;
```

		while(tx_flag == 0x00); //wait for transmit flag to set tx_flag = 0x00; //clear the flag
		}
		}
		
		void uart_interrupt(void)_irq
		{
		temp = U0IIR;
		temp = temp & 0x06; //check bits, data sending or receiving
		{
		tx_flag = 0xff; // flag that indicate data is sending via UART0 VICVectAddr=0;
		}
		
		else if(temp == 0x04) // check any data available to receive
		{
		// U0THR = U0RBR;
		emp1 = U0RBR ; // copy data into variable
		rx_flag = 0xff; // set flag to indicate that data is received VICVectAddr=0;
		}
		}

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
	Observation Table, Look-up Table, Output	
	Sample Calculations	
10	Graphs, Outputs	
11	Results &Analysis	
12	Application Areas	ALP to interface with external I/O devices is used to design interfacing solutions of computers with external devices
13	Remarks	
14	Faculty Signature with Date	

Experiment 10: .....

-	**Experiment No.:**	10	**Marks**		**Date**		**Date**	

					Planned		Conducted	
1	Title	Interface and Control a DC Motor.						
2	Course Outcomes	Able to Design and develop ARM assembly programs to interface with external I/O devices						
3	Aim	Interface and Control a DC Motor						
4	Material / Equipment Required							
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	#in
		clu
		de
		<lp
		c2
		14
		x.h
		>
		voi
		d
		clo
		ck
		_w
		ise
		(vo
		id);
		void
		anti_clo
		ck_wise
		(void);
		unsigne
		d int
		j=0;
		
		int main()
		{
		PINSEL2 = 0XFFFFFFF0;
		//IO1CLR = 0X0000ff00;
		IO1DIR= 0X00030000; //p1.16 and
		p1.17 are selected as outputs. IO1SET=
		0X00010000; //P1.16 should always high.
		
		while(1)
		{
		
		clock_wise();						

```
            for(j=0;j<500000;j++);        //delay

            anti_clock_wise();
            for(j=0;j<500000;j++);        //delay

            }                             //End of while(1)
}                                         //End of Main

void clock_wise(void)
{
            for(j=0;j<500000;j++);//small delay to allow motor to turn off
            IO1SET = 0X00030000; //Selecting the P1.17 line for clockwise
            and turn on motor
 }

void anti_clock_wise(void)
{
IO1CLR = 0X00030000;                      //stop motor and also turn off
relay
 for(j=0;j<1000000;j++);                  //small delay to allow motor to
 turn off
 IO1SET = 0X00010000;                     //not selecting the P1.17 line for
 Anti clockwise


}
```

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results &Analysis	
12	Application Areas	8086 ALP to interface with external I/O devices is used to design interfacing solutions of computers with external devices
13	Remarks	
14	Faculty Signature with Date	

Experiment 11: .....

-	**Experiment No.:**	11	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.						
2	Course Outcomes	Able to Design and develop 8086assembly programs to interface with external I/O devices						
3	Aim	Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.						
4	Material / Equipment Required							
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	#include <LPC21xx.h> void clock_wise(void) ;
		void
		anti_cloc
		k_wise(v
		oid) ;
		unsigned
		int var1 ;
		unsigned long int i = 0 , j = 0 , k = 0 ;
		
		int main(void)
		{
		PINSEL2 = 0x00000000; //P1.20 to P1.23 GPIO
		IO1DIR |= 0x00F00000 ; //P1.20 to P1.23 made as output
		
		while(1)
		{
		
		for(j = 0 ; j < 50 ; j++) // 50 times in Clock
		wise Rotation clock_wise() ; // rotate one
		round clockwise
		IO1CLR =0x00F00000 ;
		
		//clearing all 4 bits while(1);
		for(k = 0 ; k < 65000 ; k++) ; // Delay to
		show anti_clock Rotation for(j=0 ; j < 50 ; j++) // 50
		times in Anti Clock wise Rotation anti_clock_wise() ;
		// rotate one round
		anticlockwise						

		for(k = 0 ; k < 65000 ; k++) ; // Delay to show ANTI_clock Rotation
		
		}
		
		} // End of main

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results &Analysis	
12	Application Areas	interface with external I/O devices is used to design interfacing solutions of computers with external devices
13	Remarks	
14	Faculty Signature with Date	

Experiment 12: .....

-	**Experiment No.:**	12	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Determine Digital output for a given Analog input using Internal ADC of ARM controller.						
2	Course Outcomes	Able to Design and develop 8086assembly programs to interface with external I/O devices						
3	Aim	Determine Digital output for a given Analog input using Internal ADC of ARM controller.						
4	Material / Equipment Required							
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	#include <lpc214x.h> #include <Stdio.h> #define vol 3.3 // Reference voltage #define fullscale 0x3ff						

```
//10 bit adc fullscale
unsigned int
data_lcd=0,i=0,n=0;
unsigned int
adc_value=0,temp_adc=0,temp1,temp2,
adc[8]; float temp,adc1[8];
unsigned char var[15],var1[15],fst_flag=0xff; unsigned char *ptr,arr[]= "ADC O/P= ";
unsigned char *ptr1,dis[]="A I/P = ";nit(void); void wr_cn(void); void clr_disp(void);
void
delay
int main()
{
        PINSEL1 = 0X04000000;
                        //AD0.2 pin is
  selected IO0DIR = 0x000000FC;
                        //configure o/p
  lines for lcd

      delay(3200);
      lcd_init();                   //LCD initialization
 delay(3200);
      clr_disp();                   //clear display
      delay(3200);          //delay

      ptr = dis;
      temp1 = 0x80;         //Display starting address of 1st
      line on LCD lcd_com();
      delay(800);

  while(*ptr!='\0')
    ptr; lcd_data();
              ptr ++;
      }
```

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	

11	Results &Analysis	
12	Application Areas	interface with external I/O devices is used to design interfacing solutions of computers with external devices
13	Remarks	
14	Faculty Signature with Date	

Note : Remove “Table of Content” before including in CP Book
Each Course Plan shall be printed and made into a book with cover page
Blooms Level in all sections match with A.2, only if you plan to teach / learn at higher levels the uses