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

18ECL47: MICROCONTROLLER LAB

A. LABORATORY INFORMATION

1. Lab Overview

Degree:	B.E	*Program:*	EC
Year / Semester :	2 / 4	*Academic Year:*	2019-20
Course Title:	Microcontroller Lab	*Course Code:*	18ECL47
Credit / L-T-P:	2/ 2-0-0	*SEE Duration:*	180 Minutes
Total Contact Hours:	42 Hrs	*SEE Marks:*	75 Marks
CIA Marks:	40	*Assignment*	1 / Module
Course Plan Author:	Syeda N	*Sign*	Dt :
Checked By:		*Sign*	Dt :

2. Lab Content

Unit	Title of the Experiments	Lab Hours	Concept	Blooms Level
1	Data transfer – Program for block data movement, sorting, exchanging, finding largest element in an array.	3	Data Transfer	L4 Analyze
2	Arithmetic instructions: Addition, subtraction, multiplication and division. Square and cube operations for 16 bit numbers.	3	Arithmatic Operations	L4
3	Counters	3	Counters	L4
4	Boolean and logical instructions (bit manipulation).	3	Bit Manipulation Operatins	L4
5	Conditional call and return instructions.	3	Subroutine	L4
6	Code conversion programs – BCD to ASCII, ASCII to BCD, ASCII to decimal, Decimal to ASCII, Hexa decimal to and Decimal to hexa.	3	Code Conversion	L4
7	Programs to generate delay, Programs using serial port and on-chip timer/counters.	3	Timer & Serial Communication	L4
8	Stepper motor interface.	3	Peripheral Interface	L4
9	DC motor interface for direction and speed control using PWM.	3	Peripheral Interface	L4
10	Alphanumerical LCD panel interface.	3	Peripheral Interface	L4
11	Generate different waveforms: Sine, Square, Triangular, Ramp using DAC interface.	3	Peripheral Interface	L4

12	External ADC and Temperature control interface.	3	Peripheral Interface	L4
13	Elevator interface.	3	Peripheral Interface	L4

3. Lab Material

Unit	Details	Available
1	Text books	
	The 8051 Microcontroller and Embedded Systems Using Assembly and C 8051 Muhammad Ali Mazadi Pearson 2 nd Edition, 2008.	In Lib and dept
2	Reference books	
	The 8051 Microcontroller Kenneth Ayala Cengage Learning 3 rd Edition, 2005	In Lib and dept
	The 8051 Microcontroller and EmbeddedSystems Manish K Patel McGraw Hill 2014	In Lib
	Microcontrollers: Architecture,Programming, Interfacing and System Design Raj Kamal Pearson 1 st Edition, 2012	In Lib
3	Others (Web, Video, Simulation, Notes etc.)	
	VTU elearning	Not Available
	Nptel.ac.in for videoes	Not Available

4. Lab Prerequisites:

-	-	Base Course:		-	-
SNo	Course Code	Course Name	Topic / Description	Sem	Remarks
1	17ELN14/24	Basic Electronics	4/Microcontroller/Architecture and stepper motor Intereface	1/2	

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	Remarks
1	Start computer	
2	Open the Keil software	
3	Create new project	
4	Select new file.	
5	Write the program	
6	Save the program with .am or .c extension.	
7	Assemble/ Compile the program F9	
8	Execute the program F10	

B. OBE PARAMETERS

1. Lab / Course Outcomes

#	COs	Teach. Hours	Concept	Instr Method	Assessment Method	Blooms ' Level
1	Develop the program for moving data within memory locations in assembly language	10	Data Transfer	Lecture and executio n	Test and Viva	L4
2	Develop the program for additin,substraction,multiplication,division, square and root in assembly language	06	Arithmatic Operations	Lecture and executio n	Test and Viva	L4
3	Develop the program for UP/DOWN Counters in assembly language	07	Counters	Lecture and executio n	Test and Viva	L4

4	Develop the program for logical and boolean operations in assembly language	03	Bit Manipulation Operatins	Lecture and execution	Test and Viva	L4
5	Develop the program to call subroutine within main routine in assembly language	03	Subroutine	Lecture and execution	Test and Viva	L4
6	Develop the program for BCD to ASCII, ASCII to BCD, ASCII to decimal, Decimal to ASCII, Hexa decimal to and Decimal to hexa in assembly language	03	Code Conversion	Lecture and execution	Test and Viva	L4
7	Develop the program to generate time delay and serial communication using assembly language.	03	Timer & Serial Communication	Lecture and execution	Test and Viva	L4
8	Develop the program to interface stepper motor with 8051 using hardware boards.	03	Peripheral Interface	Demonstrate	Test and Viva	L4
9	Develop the program to interface DC motor with 8051 using hardware boards.	03	Peripheral Interface	Demonstrate	Test and Viva	L4
10	Develop the C program to interface LCD PANEL with 8051 using hardware boards.	03	Peripheral Interface	Demonstrate	Test and Viva	L4
11	Develop the C program to interface DAC with 8051 to generate sine, square, triangular and ramp waveforms using hardware boards.	03	Peripheral Interface	Demonstrate	Test and Viva	L4
12	Develop the C program to interface ADC with 8051 for temperature control using hardware board.	03	Peripheral Interface	Demonstrate	Test and Viva	L4
13	Develop the C program to interface elevator with 8051 using hardware board.	03	Peripheral Interface	Demonstrate	Test and Viva	L4

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

2. Lab Applications

SNo	Application Area	CO	Level
1	Data move between processor and peripheral devices.	CO1	L4
2	Calculators	CO2	L4
3	Generate PWM signal to control speed of motor or to count extrenal	CO1	L4

	events .		
4	Bit masking. Code conversion(logic & rotate instruction) & serial devices.	CO2	L4
5	All Programming languages.	CO1	L4
6	Keyboard, Printers and monitors (BCD to ASCII).	CO2	L4
7	Use for generating precise time delays in many electronic equipment such as CPU, washing Machine and microwave oven. Long distance communication.	CO1	L4
8	Automation systems, image scanners, computer printers and disc drivers.	CO3	L4
9	Elevators, air compressor, vaccum cleaner and hair driver etc	CO4	L4
10	Bank, Bus, digital watch, TV etc	CO3	L4
11	ECG Machines	CO4	L4
12	Computer, mobile and in all data acquisition systems.	CO3	L4
13	In all multistorage building .	CO4	L4

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
18EC46.1	Develop the program for moving data within memory locations in assembly language	**3**	**2**		**2**	**2**					**1**		1	L2
18EC46.2	Develop the program for additin,substraction,multiplication,division, square and root in assembly language	**3**	**2**	**2**	**1**	**1**					**1**		1	L4
18EC46.3	Develop the program for UP/DOWN Counters in assembly language	**3**	**2**		**2**	**2**					**1**		1	L2
18EC46.4	Develop the program for logical and boolean operations in assembly language	**3**	**2**	**2**	**1**	**1**					**1**		1	L4
18EC46.5	Develop the program to call subroutine within main routine in assembly language	**3**	**2**		**2**	**2**					**1**		1	L2
18EC46.6	**3**	**2**		**2**	**2**					**1**		1	L2	L4
18EC46.7	**3**	**2**	**2**	**1**	**1**					**1**		1	L4	L4

18EC46.8	Develop the program to interface stepper motor with 8051 using hardware boards.	**3**	**3**	**3**	**2**						**1**		1	L4
18EC46.9	Develop the program to interface DC motor with 8051 using hardware boards.	**3**	**3**	**3**	**3**	**2**	**1**	**1**	**2**	**1**	**2**	1	2	L4
18EC46.10	Develop the C program to interface LCD PANEL with 8051 using hardware boards.	**3**	**3**	**3**	**2**						**1**		1	L4
18EC46.11	Develop the C program to interface DAC with 8051 to generate sine, square, triangular and ramp waveforms using hardware boards.	**3**	**3**	**3**	**3**	**2**	**1**	**1**	**2**	**1**	**2**	1	2	L4
18EC46.12	Develop the C program to interface ADC with 8051 for temperature control using hardware board.	**3**	**3**	**3**	**2**						**1**		1	L4
18EC46.13	Develop the C program to interface elevator with 8051 using hardware board.	**3**	**3**	**3**	**3**	**2**	**1**	**1**	**2**	**1**	**2**	1	2	L4
18EC46.	Average													

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

4. 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.

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	Data Transfer	03	1	-	-	-	-	-	1	CO1	L4
2	Arithmatic Instructions	03	1	-	-	-	-	-	1	CO2	L4

Dept EC

Prepared by Checked by Approved

3	Counters	03	1	-	-	-	-	-	1	CO3	L4
4	Logical Instructions	03	1	-	-	-	-	-	1	CO4	L4
5	Subroutines	03	1	-	-	-	-	-	1	CO5	L4
6	Code Conversion	03	1	-	-	-	-	-	1	CO6	L4
7	Timers and Serial Communication	03	1	-	-	-	-	-	1	CO7	L4
8	Stepper Motor Interface	03	-	1	-	-	-	-	1	CO8	L4
9	DC Motor Interface	03	-	1	-	-	-	-	1	CO9	L4
10	LCD Interface	03	-	1	-	-	-	-	1	CO10	L4
11	DAC Interface	03	-	1	-	-	-	-	1	CO11	L4
12	ADC Interface	03	-	1	-	-	-	-	1	CO12	L4
13	Elevator Interface	03	-	1	-	-	-	-	1	CO13	L4
-	**Total**	**60**	**7**	**8**	**5**	**5**	**5**	**5**	**20**	-	-

Note: Write CO based on the theory course.

2. Continuous Internal Assessment (CIA)

Evaluation	Weightage in Marks	CO	Levels
CIA Exam – 1	30	CO1, CO2	L2
CIA Exam – 2	30	CO3,C04	L4
CIA Exam – 3	30	CO4	L4
Assignment - 1	05	CO1, CO2	L2
Assignment - 2	05	CO3,C04	L4
Assignment - 3	05	CO4	L4
Other Activities – define – Slip test		CO1 to CO4	L2, L3, L4 . ..
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	20 Marks
4	Internal Assessment	40 Marks
5	SEE	60Marks
-	**Total**	**100 Marks**

D. EXPERIMENTS

Experiment 01 : Data Transfer

-	**Experiment No.:**	1	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Data Transfer						
2	Course Outcomes	Develop the program for moving data within memory locations in assembly language						
3	Aim	Exercise on data transfer instructions.						
4	Material / Equipment Required	Computer ,kiel software						

Dept EC
Prepared by Checked by Approved

5	Theory, Formula, Principle, Concept	

6	Procedure, Program, Activity, Algorithm, Pseudo Code	•step 1: start
		•step 2: write programming
		•step 3: save the program
		•step 4: assemble
		•step 5:if error then correct the errors
		•step 6:run
		•step 7:stop
		
		1. ALP to transfer n-bytes of data from location x to location y without overlap.
		ORG 0000H
		MOV R3,#04H
		MOV R0,#30H
		MOV R1,#40H
		AGAIN:MOV A,@R0
		MOV @R1,A
		INC R0
		INC R1
		DJNZ R3,AGAIN
		END
		
		2.ALP to exchange n-bytes of data between location x and location y.(without using XCH)
		ORG 0000H
		MOV R3,#04H
		MOV R0,#30H
		MOV R1,#40H
		AGAIN:MOV A,@R0
		MOV 60H,@R1
		MOV @R0,60H
		MOV @R1,A
		INC R0
		INC R1
		DJNZ R3,AGAIN
		END
		
		3. ALP to exchange n-bytes of data between the location x and location y(with using XCH)
		ORG 0000H
		MOV R3,#04H

```
MOV       R0,#30H
MOV       R1,#40H
AGAIN:MOV      A,@R0
XCH       A,@R1
MOV       @R0,A
INC        R0
INC       R1
DJNZ      R3, AGAIN
END
```

4. ALP to find the largest in an array of numbers

```
            ORG       0000H
MOV       R2,#03H
            MOV       R0,#40H
RPT:  MOV         A,@R0
            INC        R0
            MOV       50H,@R0
            CJNE      A,50H,NEXT
            SJMP      NCHNGE
NEXT:  JC           NCHNGE
            XCH       A,@R0
            DEC        R0
            MOV       @R0,50H
                       INC       R0
  NCHNGE:      DJNZ      R2,RPT
        END
```

5.ALP to sort the integers in ascending order using bubble sort.

```
        ORG            0000H
       MOV            R2,#04H
       LOOP2:MOV       R3,#04H
       MOV            R0,#40H
       LOOP1:MOV       A,@R0
       INC            R0
       MOV            50H,@R0
       CJNE           A, 50H, NEXT
       SJMP           NCHNGE
       NEXT: JC                       NCHNGE
       MOV            @R0,A
       DEC            R0
       MOV            @R0,50H
```

```
        INC     R0
NCHNGE: DJNZ    R3,LOOP1
        DJNZ    R2,LOOP2
        END
```

7 Block, Circuit, Model Diagram, Reaction Equation, Expected Graph

8 Observation Table, Look-up Table, Output

1.

INPUT	30H:	93	48	96	50
OUTPUT	40H:	93	48	96	50

2.

INPUT	30H:	52	83	92	21
	40H:	22	96	98	16

OUTPUT	30H:	22	96	98	16
	40H:	52	83	92	21

4

INPUT	40H:	0	4	8	5
OUTPUT	40H:	0	4	5	8

5

INPUT	40H:	3	5	4	5	8
OUTPUT	40H:	3	4	5	5	8

9 Sample Calculations

10 Graphs, Outputs

11	Results & Analysis	
12	Application Areas	Data move between processor and peripheral devices.
13	Remarks	
14	Faculty Signature with Date	

Experiment 02 : Arithmatic Instructions.

-	**Experiment No.:**	2	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	**Arithmetic Instructions.**						
2	Course Outcomes	Develop the program for additin,substraction,multiplication,division, square and root in assembly language						
3	Aim	Exercise on arithmatic instructions.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code							

1. ALP to add two 16 bit numbers.

```
 ORG  0000H
MOV R0,#00H
MOV A,40H
ADD A,42H
MOV 52H,A
MOV A,41H
ADDC          A,43H
MOV           51H,A
JNC   NCARRY
INC   R0
NCARRY:MOV          50H,R0
END
```

2. ALP to subtract two 16 bit numbers.

```
 ORG   0000H
MOV  R0,#00H
MOV  A,40H
SUBB          A,42H
MOV  52H,A
MOV  A,41H
SUBB          A,43H
MOV  51H,A
JNC     NCARRY
```

```
  INC   R0
NCARRY:MOV          50H,R0
  END
```

3.ALP to multiply two 8 bit numbers.

```
 ORG     0000H
MOV     A,40H
MOV     B,41H
MUL     AB
MOV     51H,A
MOV     50H,B
END
```

4. ALP to divide two 8 bit numbers.

```
 ORG     0000H
MOV     A,40H
MOV     B,41H
DIV     AB
MOV     51H,A
MOV     50H,B
END
```

5. ALP to find square of a 8 bit numbers.

```
 ORG     0000H
MOV     A,40H
MOV     B,40H
MUL     AB
MOV     51H,A
MOV     50H,B
END
```

7 Block, Circuit, Model Diagram, Reaction Equation, Expected Graph

1.

INPUT	40H:	D0	C7	E2	D1
OUTPUT	50H:	01	99	B2	

2

INPUT	40H:	4E	73	F2	AD
OUTPUT	50H:	01	C5	5C	

3

		INPUT 40H 96 25 OUTPUT 50H 15 AE 4 INPUT 40H 32 05 OUTPUT 50H 00 0A 5 INPUT 40H 0F OUTPUT 50H 00 E1
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	Calculators
13	Remarks	
14	Faculty Signature with Date	

Experiment 03 : Counters

-	**Experiment No.:**	3	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Counters						
2	Course Outcomes	Develop the program for UP/DOWN Counters in assembly language						
3	Aim	Exercise on DEC/INC instructions						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	To identify the key words in c programming To identify the identifiers in c programming						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. ALP for HEXADECIMAL UP/DOWN counter. ORG 0000H MOV A,#00H **LOOP**: ACALL DELAY INC A CJNE A, #0FFH, LOOP **LOOP1**:ACALL DELAY						

		DEC A
		CJNE A, #00H, LOOP1
		SJMP $
		
		DELAY::MOV R1,#0FFH
		DECR:1:MOV R2,#0FFH
		DECR: MOV R3,#0FFH
		HERE: DJNZ R3,HERE
		DJNZ R2,DECR
		DJNZ R1,DECR1
		RET
		
		2.ALP for BCD UP/DOWNcounter.
		ORG 0000H
		MOV A,#00H
		LOOP: ACALL DELAY
		ADD A,#01H
		DA A
		CJNE A ,#99H, LOOP
		LOOP1:ACALL DELAY
		ADD A,#99H
		DA A
		CJNE A , #00H, LOOP1
		SJMP $
		
		DELAY:MOV R1,#0FFH
		DECR1:MOV R2,#0FFH
		DECR: MOV R3,#0FFH
		HERE: DJNZ R3,HERE
		DJNZ R2,DECR
		DJNZ R1,DECR1
		RET

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	**OUTPUT**: 1. A: 00H.....FFH.....00H //STACK WINDOW 2. **A:** 00.....99.....00 //STACK WINDOW
9	Sample Calculations	

10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	Generate PWM signal to control speed of motor or to count extrenal events
13	Remarks	
14	Faculty Signature with Date	

.....

Experiment 04 :Logical Instructions

-	**Experiment No.:**	4	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Logical Instructions						
2	Course Outcomes	Develop the program for logical and boolean operations in assembly language						
3	Aim	Exercise on logical instructions.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	2.AND operation is used to RESET the bits, OR operation is used to SET the bits, XOR operation is used to COMPLEMENT the bits.						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	1.ALP to perform the following operations on 3 bytes of data stored from location x
		a. OR the lower nibble of x with upper nibble of x+1.
		b. RESULT of a is XOR with x+2
		
		ORG 0000H
		MOV A,40H
		ANL A,#0FH
		MOV 50H,A
		MOV A,41H
		SWAP A
		ANL A,#0FH
		ORL A,50H
		XRL A,42H
		MOV 51H,A
		END
		2. ALP to perform the following operations on a given byte of data. .SET the bits of 0 and 1,RESET the bits 2 and 3, COMPLEMENT the bits 6 and 7.
		ACCESSING BITS:
		
		ORG 0000H						

```
MOV     A,40H
SETB    0E0H
SETB    0E1H
CLR     0E2H
CLR     0E3H
CPL     0E6H
CPL     0E7H
MOV     50H,A
END
```

ACCESSING BYTE:

```
ORG     0000H
MOV     A,40H
ORL     A,#03H
ANL     A,#0F3H
XRL     A,#0C0H
MOV     50H,A
END
```

3. ALP to count number of 1's and 0's in a given byte of data.

```
ORG     0000H
MOV     R2,#00H
MOV     R3,#00H
MOV     R4,#08H
CLR     C
MOV     A,40H
RPT:  RLC A
JC      NEXT
INC     R2
SJMP    NEXT2
NEXT: INC     R3
NEXT2:DJNZ    R4,RPT
MOV     50H,R2
MOV     51H,R3
END
```

4.ALP to check whether the given byte is odd or even , if it is an even number store 00 in 50H else store EEH in 50H.

```
ORG     0000H
MOV     A, 40H
```

```
RRC     A
JC      NEXT
MOV     50H,#00H
SJMP    $
NEXT: MOV     50H,#0EEH
SJMP    $
```

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	1.

INPUT	40H	25	1B	54
OUTPUT	51H	51		

2.

INPUT	40H	54
OUTPUT	50H	93

3.

INPUT	40H	88	
OUTPUT	50H	06	02

4.

INPUT	40H	84
OUTPUT	50H	00

9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	Bit masking. Code conversion(logic & rotate instruction) & serial devices
13	Remarks	
14	Faculty Signature with Date	

.....

Experiment 05 : Subroutines

-	**Experiment No.:**	5	**Marks**	10	**Date Planned**		**Date Conducted**	

1	Title	Subroutines
2	Course Outcomes	Develop the program to call subroutine within main routine in assembly language
3	Aim	Exercise on CALL and RET instructions.
4	Material / Equipment Required	Computer ,kiel software
5	Theory, Formula, Principle, Concept	LOGIC:1. In the main program separate 2 digits into two individual single digits.**Seperation Logic:** AND the given byte with 0FH to get lower digit. Swap the given byte,then AND with 0Fh to get the upper digit. 2. For each digit ASCII Conversion, call the subroutine ASCII. **ASCII Conversion Logic:** If the digit is less then 0Ah, add 30h, If digit is greater than 0Ah, add 37h.

6	Procedure, Program, Activity, Algorithm, Pseudo Code	ALP to convert hexadecimal to ASCII.
		ORG 0000H
		MOV A,50H
		ANL A,#0FH
		ACALL ASCII
		MOV 60H,A
		MOV A,50H
		SWAP A
		ANL A,#0FH
		ACALL ASCII
		MOV 61H,A
		SJMP $
		
		ASCII: CJNE A, #0AH, NEXT
		SJMP NEXT2
		NEXT: JC NEXT1
		NEXT2:ADD A,#37H
		RET
		NEXT1:ADD A,#30H
		RET

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

INPUT	50H	FE	
OUTPUT	60H	45	46

9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	All Programming languages
13	Remarks	
14	Faculty Signature with Date	

Experiment 06 : Code Conversion

-	**Experiment No.:**	6	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Code Conversion						
2	Course Outcomes	Develop the program for BCD to ASCII, ASCII to BCD, ASCII to decimal, Decimal to ASCII, Hexa decimal to and Decimal to hexa in assembly language .						
3	Aim	Exercise on code conversion.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	LOGIC: 1&2: ASCII values hex values 30-39 0-9 41-46 A-F LOGIC:3:Divide the given number by 100d(64h),the quotient of the division is 100's position digit, divide the remainder by 10d(Ah).Quotient of second division is then ten's position digit and remainder is unit's position digit.						

	Procedure, Program, Activity, Algorithm, Pseudo Code	1.ALP to convert ASCII Code to binary(hexadecimal) Code.
		If ASCII value is less than 40h then subtract 30h to get the hexadecimal value.
		If ASCII value is greater than 40h then subtract 37h to get the hexadecimal value.
		ORG 0000H
		MOV A,50H
		CJNE A, #40H,NEXT
		SJMP FINAL
		NEXT: JC NEXT1
		SUBB A,#37H
		SJMP RESLT
		NEXT1: CLR C						

```
        SUBB    A,#30H
RESLT:MOV 60H,A
        SJMP    $
```

2. ALP to convert to BCD to hexadecimal.

```
 ORG     0000H
MOV     A,50H
ANL     A,#0FH
MOV     52H,A           //LD
MOV     A,50H
SWAP    A
ANL     A,#0FH
MOV     51H,A           // UD is stored in A as well as in
51H
MOV     B,#0AH
MUL     AB      // A = UD * 0AH
ADD     A,52H           //HEXA = (UD*0AH) + LD
MOV     53H,A
END
```

3. ALP to convert hexadecimal to BCD.

```
 ORG     0000H
MOV     A,40H
MOV     B,#100D
DIV             AB
MOV     50H,A
MOV     A,#0F0H
MOV     B,#10D
DIV             AB
MOV     51H,A
MOV      52,B
END
```

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	1.

INPUT	50H	42
OUTPUT	60H	0B

		2. INPUT 50H 99 OUTPUT 50H 99 09 09 63 3 INPUT 40H FF OUTPUT 50H 02 05 05
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	Developed and executed C program
12	Application Areas	Keyboard, Printers and monitors (BCD to ASCII).
13	Remarks	
14	Faculty Signature with Date	

Experiment 07 : Timers and Serial Communication

-	**Experiment No.:**	8	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Timers and Serial Communication						
2	Course Outcomes	Develop the program to generate time delay and serial communication using assembly language.						
3	Aim	Exercise on timers and serial communication.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	**Calculations:** f=100khz $t=10*10^{6}s$ 1 pulse=$5*10^{6}s$ $n = \frac{5*10^{6}}{1.085*10^{6}} = 4.61 = 5$ initial value=65536 – n = 65531=FFFBH.						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	1.ALP to generate a square wave of 100khz using timer0 in mod1.
		ORG 0000H
		MOV TMOD,#01H
		AGAIN: MOV TL0,#0FBH
		MOV TH0,#0FFH						

```
        CPL     P0.0
        SETB    TR0
HERE:   JNB     TF0,HERE
        CLR     TR0
        CLR     TF0
        SJMP    AGAIN
```

2.ALP to generate serial code using serial port and on-chip timer/counter.

```
        ORG     0000H
        MOV     TMOD,#20H
        MOV     TH1,#-3
        MOV     SCON,50H
        SETB    TR1
UP:     MOV     A,#'S'
        ACALL   SEND
        MOV     A,#'K'
        ACALL SEND
        MOV     A,#'I'
        ACALL   SEND
        MOVA,   #'T'
        ACALL   SEND
        MOV     A, #'  '
        SJMP    UP

SEND:   MOV     SBUF,A
HERE:   JNB     TI,HERE
        CLR     TI
        RET
```

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	1. 2.SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT SKIT
9	Sample Calculations	
10	Graphs, Outputs	

11	Results & Analysis	
12	Application Areas	Use for generating precise time delays in many electronic equipment such as CPU, washing Machine and microwave oven. Long distance communication.
13	Remarks	
14	Faculty Signature with Date	

Experiment 08 : Stepper Motor Interface

-	**Experiment No.:**	8	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Stepper Motor Interface						
2	Course Outcomes	Develop the program to interface stepper motor with 8051 using hardware boards						
3	Aim	Exercise on stepper motor interface with 8051.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	Stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The motors rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the motor shafts rotation is directly related to the frequency of the input pulses and the length of rotation is directly related to the number of input pulses applied.						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	A C-program to run the stepper motor in clockwise direction.
		#include<reg51.h>
		void delay(unsigned int);
		void main( ){
		while(1){
		P0=0x99;
		delay(1);
		P0=0x33;
		delay(1);
		P0=0x66;
		delay(1);
		P0=0xCC;
		delay(1);						

		}
		}
		void main(unsigned int value){
		unsigned int i,j;
		for (i = 0; i < 100; i++)
		for (j = 0 ;j < value; j++)
		;
		}

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	Developed and executed C program
12	Application Areas	Automation systems, image scanners, computer printers and disc drivers.
13	Remarks	
14	Faculty Signature with Date	

Experiment 09 : DC Motor Interface

-	**Experiment No.:**	9	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	DC Motor Interface						
2	Course Outcomes	Develop the program to interface DC motor with 8051 using hardware boards.						
3	Aim	Exercise on DC motor interface with 8051.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	DC motors are used in many applications like process control and automation in an industry, robotics, consumer electronics, office automation equipment like printers and scanners etc. One can consider the use of a DC motor wherever there is need to control the motion of an object. Speed control of the motor is important in the applications involving them. For example, in an audio system, the DC motor that drives the cassette should always run at a fixed speed.						

		Like wise, there are applications where the speed of the DC motor has to change according to some defined conditions. The DC motor used in this interface module is a 12V, 4W motor that can be seen in many electronic equipments. The circuit to control the speed of the motor follows a general concept and can be applied to DC motors of higher capacity also. The pulse width modulation technique is used to vary the speed of the DC motor. The frequency of the pulses is 120Hz. Keeping the frequency constant, the width of the pulses is used to change the speed. When the pulse width is minimum, the speed is minimum and when the width is maximum, the speed is maximum (2400rpm). The ramp and pedestal technique is used to change the pulse width and thereby the speed

6	Procedure, Program, Activity, Algorithm, Pseudo Code	write a C-program to control DC motor.
		#include<rreg51.h>
		#include<stdio.h>
		void delay(void);
		sbit motor_pin_1=P2^1;
		sbit motor_pin_2=P2^3;
		sbit en_bit=P2^0;
		void main( ){
		en_bit=1;
		do{
		motor_pin_1=1; // ANTICLOCKWISE
		motor_pin_2=0;
		delay( );
		motor_pin_1=1;
		motor_pin_2=1; //STOP
		delay( );
		motor_pin_1=0;
		motor_pin_2=1; // CLOCKWISE
		delay( );
		motor_pin_1=0;
		motor_pin_2=0; //STOP
		delay( );
		} while(1);
		}
		void delay( ){
		int i,j;
		for (i = 0; i < 1000; i++){
		for (j = 0; j < 1000; j++)
		;
		}

		}
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	Developed and executed C program
12	Application Areas	Elevators, air compressor, vaccum cleaner and hair driver etc
13	Remarks	
14	Faculty Signature with Date	

Experiment 10 : LCD Interface

-	**Experiment No.:**	10	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	LCD Interface						
2	Course Outcomes	Develop the C program to interface LCD PANEL with 8051 using hardware boards.						
3	Aim	Exercise on LCD interface with 8051.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	**C program to display the temperature on LCD screen** #include <Intel\8051.h> #include <standard.h> #define PORTA 0x2040 #define PORTB 0x2041 #define PORTC 0x2042 #define CNTL 0x2043 #define buff 0x196 xdata unsigned char *p8255_cntl ; xdata unsigned char *p8255_porta ; xdata unsigned char *p8255_portb ; xdata unsigned char *p8255_portc ;						

```
xdata unsigned char *buff_ptr;
idata unsigned char temp1,adc_val;
void main ()
{
buff_ptr=buff; // mem. locn to hold adc data to display
p8255_porta = PORTA;
p8255_portc = PORTC;
p8255_portb = PORTB;
p8255_cntl = CNTL;
*p8255_cntl = 0x98;// Ppa=i/p,Pb=o/p,PCu=i/p,PCl=o/p,
*p8255_cntl = 0x03;// channel 1 selection Wr=1,PC1=1
*p8255_cntl = 0x00;// start=0, PC0=0
delay(200);
while(1)
{
p8255_porta = PORTA;
p8255_portc = PORTC;
p8255_portb = PORTB;
p8255_cntl = CNTL;
*p8255_cntl = 0x01;// start=1,PC0=1
delay(200);
*p8255_cntl = 0x00;// start=0, PC0=0// check for eoc,PC7=1
do
{
temp1=*p8255_portc;
temp1=temp1 & 0x80;
} while(temp1 != 0x80);

//delay(200);// after eoc, read the adc data from PA
adc_val = *p8255_porta;// display adc result on the data field
*buff_ptr = adc_val;
// This assembly program displays the adc_val on LCD screen
ACC=*buff_ptr;
asm a,#00h
asm da a
asm mov r6,a
asm lcall 677dh
asm mov r0,0ffh
asm mov r1,0ffh
asm lcall 6850h
asm mov r0,0ffh
asm mov r1,0ffh
```

		asm lcall 6850h
		delay(200);
		} // end of while(1)
		}

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	Developed and executed C program
12	Application Areas	Bank Token display, Bus stop display , digital watch, TV etc
13	Remarks	
14	Faculty Signature with Date	

Experiment 11 : DAC Interface

-	**Experiment No.:**	11	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	DAC Interface						
2	Course Outcomes	Develop the C program to interface DAC with 8051 to generate sine, square, triangular and ramp waveforms using hardware boards.						
3	Aim	Exercise on DAC interface with 8051.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	A 8051 C program to generate rectangular wave using DAC interface.
		#include<reg51.h>
		void delay(char);
		void main( ){
		while (1){
		P2=0x00;
		delay (100);
		P2=0xFF;
		delay (50);
		}						

```
}

void delay(char value){
            unsigned int i ;
              for(i = 0; i <value; i++)
                ;
 }
```

A 8051 C program to generate square wave using DAC interface.

```
#include<reg51.h>
void delay(char);
void main( ){
    while(1){
        P2=0x00;
        delay(50);
        P2=0xFF;
        delay(50)
        }
}

void delay(char value){
            unsigned int i;
            for( i = 0; i < value; i++)
                ;
 }
```

A C-program to generate a ramp waveform using DAC interface.

```
#include<reg51.h>
void main( ){
       unsigned char i;
       while(1){
             for(i = 0; i <= 255; i++)
             P2 = i;
       }
 }
```

A C-program to generate a triangular waveform using DAC interface

in 8051.

```
#include<reg51.h>
void main( ){
        unsigned char i;
        while(1){
                for(i = 0; i < 255; i++)
                P2 = i;
                for(i = 255; i > 0; i--)
                P2 = i;
        }
}
```

A program to generate sine wave using DAC interface in 8051.

```
#include<reg51.h>
void main( ){

                        unsigned int i;
                                                unsigned char
table[13]={128,192,238,255,238,192,128,64,17,0,17,64,128};
                        while(1){
                for ( i = 0; i < 13; i++)
                P0 = table[i];
          }
}
```

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	Developed and executed C program
12	Application Areas	ECG Machines
13	Remarks	
14	Faculty Signature with Date	

Experiment 12 : ADC Interface

-	Experiment No.:	12	Marks	10	Date Planned		Date Conducted	
1	Title	ADC Interface						
2	Course Outcomes	Develop the C program to interface ADC with 8051 for temperature control using hardware board.						
3	Aim	Exercise on ADC interface with 8051.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept							

6	Procedure, Program, Activity, Algorithm, Pseudo Code	#include <reg51xd2.h>
		#include "lcd.h"
		unsigned int Adc;
		unsigned char Low_adc,High_adc,relay;
		read_adc()
		{
		unsigned char status;
		
		P2_3 = 1 ; // Start conversion of ADC
		status = P1; //Read status of ADC
		while((status & 0x01) != 0x01)
		{ status = P1;
		}
		P2_2 = 0; // Enable outputs
		P2_0 = 0; // Activate B1 to B8 outputs
		Low_adc = P0; // Read lower byte of ADC and place in R0
		P2_0 = 1; // Deactivate B1 to B8 outputs
		P2_1 = 0; // Activate B9 to B12 and POL, over range outputs
		High_adc = P0; // Read higher byte of ADC
		High_adc = High_adc & 0x0F;
		P2_1 = 1; // deactivate B9 to B12 and POL, over range outputs
		
		P2_2 = 1; // Disable outputs
		P2_3 = 0; // Stop conversion of ADC
		}
		
		main() {
		float Temp,Vol,Res;						

```
unsigned char Temp1;
unsigned char Temp2,Temp3; P0 = 0xFF ; // Make port 0 as input
P2 = 0xFF ; // Make port 2 as high now the relay is on.
 P1_1 = 0 ; // switch OFF relay
 P2_3 = 0 ; // STOP conversion of ADC
 relay = 10;
while(1) {
read_adc(); //Read ADC
Adc = High_adc;
Adc <<= 8;
 Adc = Adc | Low_adc;
if( (Adc > 0x656) && (relay != 0)) //IF greater than 0x0656 Switch OFF
relay
{
 ClrLcd();
WriteString("RELAY OFF");
 P1_1 = 0 ;
relay = 0;
}
else if ( (Adc < 0x5b9) && (relay!= 1)) //IF less than 0x05B9 Switch
ON relay
{
ClrLcd();
WriteString("RELAY ON");
P1_1 = 1 ; relay = 1;
 }
Vol =-((Adc/10)*0.000488); //voltage before amplifier
Res  =((100*(1.8-Vol)-100*Vol)*100)  /(100*Vol  +  100*(1.8+Vol));//
Resistance Value Res = Res - 100; Temp = Res/ 0.384;
Temp1 = Temp;
Temp2 = 0x30 + (Temp1 / 0x0A);
Temp3 = 0x30 + (Temp1 % 0x0A);
GotoXY(0,1);
WriteString("Temperature ");
WriteChar(Temp2);
 WriteChar(Temp3);
 WriteString("'C");
}
 }
```

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	Developed and executed C program to interface ADC with 8051 for temperature control using hardware board.
12	Application Areas	Computer, mobile and in all data acquisition systems
13	Remarks	
14	Faculty Signature with Date	

Experiment 13 : Elevator Interface

-	**Experiment No.:**	13	**Marks**	10	**Date Planned**		**Date Conducted**	
1	Title	Elevator Interface						
2	Course Outcomes	Develop the C program to interface elevator with 8051 using hardware board.						
3	Aim	Exercise on elevator interface with 8051.						
4	Material / Equipment Required	Computer ,kiel software						
5	Theory, Formula, Principle, Concept	This interface has four keys, marked 0, 1, 2, and 3 representing the request buttons at the four floors. Pressing of key causes a corresponding Flip-Flop to be set. The outputs of the four Flip-flops can be read through port B (PBO, PBI, PB2 and PB3). Also, the status of these signals is reflected by a setoff 4 LEDs. The Flip-Flop can be rest (LEDs are cleared) through port A (PA54, PA5, PA6, and PA7). A column of 10 LEDs, representing the elevator can be controlled through Port A (PA0, PA1, PA2 and PA3). These port lines are fed to the inputs of the decoder 7442 whose outputs are used to control the on/off states of the LEDs which simulate the motion of the elevator.						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	#include <reg51xd2.h>
		void delay(unsigned int);
		main() {
		unsigned char Flr[9]={0xff,0x00,0x03,0xff,0x06,0xff,0xff,0xff,0x09};
		unsigned char FClr[9]={0xff,0x0E0,0x0D3,0xff,0x0B6,0xff,0xff,0xff,0x79};
		unsigned char ReqFlr,CurFlr = 0x01,i,j;						

```
P0 = 0x00; P0 = 0x0f0;
while(1) {
 P1 = 0x0f;
ReqFlr = P1 | 0x0f0;
while(ReqFlr == 0x0ff)
ReqFlr = P1 | 0x0f0; //Read Request Floor from P1
ReqFlr = ~ReqFlr;
if(CurFlr == ReqFlr) //If Request floor is equal to Current Floor
{
 P0 = FClr[CurFlr]; //Clear Floor Indicator
 continue; //Go up to read again
 }
else if(CurFlr > ReqFlr) //If Current floor is > request floor
 {
i = Flr[CurFlr] - Flr[ReqFlr]; //Get the no of floors to travel
j = Flr[CurFlr]; for(;i>0;i--) // Move the indicator down
 {
P0 = 0x0f0|j;
 j--;
delay(50000);
 }
 }
else // If Current floor is < request floor
{ i = Flr[ReqFlr] - Flr[CurFlr]; //Get the no of floors to travel
 j = Flr[CurFlr];
 for(;i>0;i--) // Move the indicator Up
 { P0 = 0x0f0 | j;
j++;'
 delay(50000);
}
 }
 CurFlr = ReqFlr; // Update Current floor

P0 = FClr[CurFlr]; // Clear the indicator
 }
 }
void delay(unsigned int x)
 {
 for(;x>0;x--);
 }
```

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	Developed and executed C program to interface elevator with 8051 using hardware board.
12	Application Areas	In all multistorage building
13	Remarks	
14	Faculty Signature with Date	