QMP 7.1 D/F



Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015 Certified Institution)



NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka

**Department of Electronics & Communication Engineering** 

### MICROCONTROLLERS LABORATORY BECL456A CIT GUBBL/Enger DEPT

Lab Manual 2024-25

Name :	
USN :	
Batch :	_Section :



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015 Certified Institution)



NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka

**Department of Electronics & Communication Engineering** 

### MICROCONTROLLERS LABORATORY BECL456A

## **CIT GUBBI ECE DEPT**

**Prepared by:** Vinaya Kumar S R & Sowmya M R Assistant Professor **Reviewed by:** Vinaya Kumar S R Assistant professor

**Approved by:** Dr.Thejaswini R

Professor & Head,Dept. of ECE





(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015 Certified Institution)

### **INSTITUTE VISION**

To create centres of excellence in education and to serve the society by enhancing the quality of life through value based professional leadership.

### **INSTITUTE MISSION**

- To provide high quality technical and professionally relevant education in a diverse learning environment.
- To provide the values that prepare students to lead their lives with personal integrity, professional ethics and civic responsibility in a global society.
- To prepare the next generation of skilled professionals to successfully compete in



all races, creeds and cultures, values and intellectual curiosity, pursuit of knowledge and academic integrity and freedom.

- To offer a wide variety of off-campus education and training programmes to individuals and groups.
- To stimulate collaborative efforts with industry, universities, government and professional societies.
- To facilitate public understanding of technical issues and achieve excellence in the operations of the institute.

### **QUALITY POLICY**

Our organization delights customers (students, parents and society) by providing value added quality education to meet the national and international requirements. We also provide necessary steps to train the students for placement and continue to improve our methods of education to the students through effective quality management system, quality policy and quality objectives.



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015 Certified Institution)

### Vision

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

To create globally competent Electronics and Communication Engineering

### **Mission**

professionals with ethical and moral values for the betterment of the society

 To nurture the technical/professional/engineering and entrepreneurial skills for overall self and societal upliftment through co-curricular and extra-curricular events.



- To create the Centres of Excellence in the field of electronics and communication in collaboration with industries/Universities by training the faculty through latest technologies.
- To impart quality technical education in the field of electronics and communication engineering to meet over the current/future global industry requirements.



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015Certified Institution) NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka



### Program Educational Objectives (PEO's)

After four Years of Graduation, our graduates are able to:

- Provide technical solutions to real world problems in the areas of electronics and communication by developing suitable systems.
- Pursue engineering career in Industry and/or pursue higher education and research.
- Acquire and follow best professional and ethical practices in Industry and Society.
- Communicate effectively and have the ability to work in term and to lead the term. I GUBB ECE DEPT

### **Program Specific Outcomes (PSO's)**

### At the time of graduation, our graduates are able to:

**PSO1:** Specify, design, build and test analog and digital systems for signal processing including multimedia applications, using suitable components or Simulation tools.

**PSO2:** Understand and architect wired and wireless analog and digital Communication systems as per specifications and determine their performance.



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015Certified Institution) NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

### SYLLABUS

	B.E: Electronics & Comn NEP, Outcome	unication Engineering / B.E: Ele Based Education (OBE) and Ch (Effective from the a SEMES	ectronics & Telecommunication Engineering noice Based Credit System (CBCS) cademic year 2021 – 22) STER – IV	
	En	bedded C Basics L	ABORATORY	
Laborat	tory Code	BECL456A	CIE Marks	50
Number	lumber of Lecture Hours/Week 0:0:2:0 SEE Marks			
RBT Level L2, L3 Exam Hours				03
		CREDITS -	01	
		Laboratory Exp	eriments	
	<b>CIT G</b>	JBBL	ECE DE	<b>PT</b>
SI.No		Expe	eriments	•
	Execute the following exp be chosen as the target)	eriments by using Keil M and Hardware Interfacin	licrovision Simulator (any 8051 Mic g Programs using 8051 Trainer Kit.	rocontroller can
		Assembly Lang	uage Programming	
1	Write an ALP to move a bl RAM.	ock of n bytes of data fro	om source (20h) to destination (40h)	using Internal-
2	Write an ALP to move a bl External RAM.	ock of n bytes of data fro	om source (2000h) to destination (20	)50h) using
3	Write an ALP To exchange	the source block starting	g with address 20h, (Internal RAM) c	ontaining N (05)
-	bytes of data with destina	tion block starting with a	address 40h (Internal RAM).	
4	Write an ALP to exchange	the source block starting	g with address 10h (Internal memory	v), containing n
	(06) bytes of data with de	stination block starting a	t location 00h (External memory).	
5	Write an ALP to add the b (MSB), using Indirect Addr	yte in the RAM at 34h an ressing Mode.	d 35h, store the result in the registe	r R5 (LSB) and R6
6	Write an ALP to subtract t	he bytes in Internal RAM	34h &35h store the result in registe	er R5 (LSB) & R6
	(MSB).			
7	Write an ALP to multiply t	wo 8-bit numbers stored	at 30h and 31h and store16- bit res	ult in 32h and 33h
	of Internal RAM.			
8	write an ALP to perform c	ivision operation on 8-bi	it number by 8-bit number.	
9	write an ALP to separate	positive and negative in a	a given array.	
10	write an ALP to separate of	even or odd elements in	a given array.	
11	write an ALP to arrange th	ne numbers in Ascending	K Descending order.	
12	write an ALP to find Large	st & Smallest number fro	om a given array starting from 20h &	store it in
10				
13	Write an ALP for Decimal	DOWN Counter		
14	Write an ALP for Lloyeder			
15	write an ALP for Hexadec	mai UP-Counter.		

16	Write an ALP for Hexadecimal DOWN-Counter.			
	C Programming			
17	Write an 8051 C program to find the sum of first 10 Integer Numbers.			
18	Write an 8051 C program to find Factorial of a given number.			
19	Write an 8051 C program to find the Square of a number (1 to 10) using Look-Up Table.			
20	Write an 8051 C program to count the number of Ones and Zeros in two consecutive memory locations.			
Hardware Interfacing Programs				
21	Write an 8051 C Program to rotate stepper motor in Clock & Anti-Clockwise direction.			
22	Write an 8051 C program to Generate Sine & Square waveforms using DAC interface.			

#### **Conduct of Practical Examination:**

- All laboratory experiments are to be included for practical examination.
- Students are allowed to pick one experiment from the lot.
- Strictly follow the instructions as printed on the cover page of answer script for breakup of marks.
- Change of experiment is allowed only once and Marks allotted to the procedure part to be made zero.



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015Certified Institution) NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka



### **Department of Electronics and Communication Engineering**

\_\_\_\_\_

### ----

**Embedded C LABORATORY: Course Objectives and Outcomes** 

### **COURSE OBJECTIVES :**

### The main objectives of this lab are, **CITGUBBIECE DEPT** • Understand the basic programming of Microcontrollers.

- Developthe8051 Microcontroller-basedprogramsforvariousapplicationsusing Assembly Language & C Programming.
- Program8051MicrocontrollertocontrolanexternalhardwareusingsuitableI/Oports.

### **COURSE OUTCOMES :**

### After completing this course the student could be able to:

\_\_\_\_\_

- 1. Write a Assembly Language/C programs in 8051 for solving simple problems that manipulate input data using different instructions.
- 2. Write C programs in 8051 for solving simple problems that manipulate input data using different instructions of 8051 C.
- 3. Develop testing and experimental procedures on 8051 Microcontroller, analyze their operation under different cases.
- 4. Develop programs for 8051 Microcontroller to implement real world problems.
- 5. Develop microcontroller applications using external hardware inter



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015Certified Institution) NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka



Department of Electronics and Communication Engineering

### <u>'Instructions to the Candidates'</u>

- Student should come with thorough preparation for the experiment to be conducted.
- Student should take prior permission from the concerned faculty before availing the leave.
- Student should come with proper dress code and to be present on time in the laboratory.
- Student will not be permitted to attend the laboratory unless they bring the practical record fully completed in all respects pertaining to the experiment conducted in the previous class.

Student will not be permitted to attend the laboratory unless they bring the observation book fully completed in all respects pertaining to the experiment to be conducted in present class.

- Experiment should be started conducting only after the staff-in-charge has checked the circuit diagram.
- All the calculations should be made in the observation book. Specimen calculations for one set of readings have to be shown in the practical record.
- Wherever graphs to be drawn, A-4 size graphs only should be used and the same should be firmly attached in the practical record.
- Practical record and observation book should be neatly maintained.
- Student should obtain the signature of the staff-in-charge in the observation book after completing each experiment.
- Theory related to each experiment should be written in the practical record before procedure in your own words with appropriate references.



(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015Certified Institution) NH 206 (B.H. Road), Gubbi, Tumkur – 572 216.

Karnataka



### **Department of Electronics and Communication Engineering**

CONTENTS					
SI.No.	Name of the Experiment	Page No.			
1.	Write an ALP to move a block of n bytes of data from source (20h) to destination (40h) using Internal-RAM.	17			
2.	Write an ALP to move a block of n bytes of data from source (2000h) to destination (2050h) using External RAM.	18			
3.	Write an ALP To exchange the source block starting with address 20h, (Internal RAM) containing N (05) bytes of data with destination block starting with address 40h (Internal RAM).	19			
4.	Write an ALP to exchange the source block starting with address 10h (Internal memory), containing n (06) bytes of data with destination block starting at location 00h (External memory).	20			
5.	Write an ALP to add the byte in the RAM at 34h and 35h, store the result in the register R5 (LSB) and R6 (MSB), using Indirect Addressing Mode.	21			
6.	Write an ALP to subtract the bytes in Internal RAM 34h &35h store the result in register R5 (LSB) & R6 (MSB).	22			
7.	Write an ALP to multiply two 8-bit numbers stored at 30h and 31h and	23			
8.	Wite in ALP operform in signation on 8-bit number by 8-bit number.	24			
9.	Write an ALP to separate positive and negative in a given array.	25			
10.	Write an ALP to separate even or odd elements in a given array.	26			
11.	Write an ALP to arrange the numbers in Ascending & Descending order.	27			
12.	Write an ALP to find Largest & Smallest number from a given array starting from 20h & store it in Internal Memory location 40h.	28			
13	Write an ALP for Decimal UP-Counter.	29			
14	Write an ALP for Decimal DOWN-Counter.	30			
15	Write an ALP for Hexadecimal UP-Counter.	31			
16	Write an ALP for Hexadecimal DOWN-Counter.	32			
17	Write an 8051 C program to find the sum of first 10 Integer Numbers.	33			
18	Write an 8051 C program to find Factorial of a given number.	34			
19	Write an 8051 C program to find the Square of a number (1 to 10) using Look-Up Table.	35			
20	Write an 8051 C program to count the number of Ones and Zeros in two consecutive memory locations.	36			
21	Write an 8051 C Program to rotate stepper motor in Clock & Anti- Clockwise direction.	37			
22	Write an 8051 C program to Generate Sine & Square waveforms using DAC interface.	39			



### Channabasaveshwara Institute ofTechnology (Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (NAAC Accredited & ISO 9001:2015Certified Institution) NH 206 (B.H. Road), Gubbi, Tumkur – 572 216.



Karnataka

### Department of Electronics and Communication Engineering

### **INDEX PAGE**

SI. No	Name of the Experiment		Date	l Marks x . 20)	l Marks IX. 10)	ature dent)	ature culty)	
		Conduction	Repetition	Submission of Record	Manua (Ma	Record (Ma	Sign (Stu	Sign (Fae
1	Write an ALP to move a block of n bytes of data from source (20h) to destination (40h) using Internal-RAM.							
2	Write an ALP to move a block of n bytes of data from source (2000h) to destination (2050h) using External RAM.							
3	Write an ALP To exchange the source block starting with addross 20h (Internal PAM) containing N (45) wytes of data with the destination block starting with address 40h (Internal RAM).	BIE	ECI	E DI	EF			
4	Write an ALP to exchange the source block starting with address 10h (Internal memory), containing n (06) bytes of data with destination block starting at location 00h (External memory).							
5	Write an ALP to add the byte in the RAM at 34h and 35h, store the result in the register R5 (LSB) and R6 (MSB), using Indirect Addressing Mode.							
6	Write an ALP to subtract the bytes in Internal RAM 34h &35h store the result in register R5 (LSB) & R6 (MSB).							
7	Write an ALP to multiply two 8-bit numbers stored at 30h and 31h and store16- bit result in 32h and 33h of Internal RAM.							

8	Write an ALP to perform division operation on 8-bit number by 8-bit number.					
9	Write an ALP to separate positive and negative in a given array.					
10	Write an ALP to separate even or odd elements in a given array.					
11	Write an ALP to arrange the numbers in Ascending & Descending order.					
12	Write an ALP to find Largest & Smallest number from a given array starting from 20h & store it in Internal Memory location 40h. Write an ALP for Decimal UP-Counter.					
13 14	Write an ALP for Decimal DOWN-Counter.					
15	Write an ALP for Hexadecimal UP-Counter.					
16	Write an ALP for Hexadecimal DOWN- Counter	<b>BIF</b>	FCI	= DI		
17	Write an 3051-C plogram to and the sum of first 10 Integer Numbers.					
18	Write an 8051 C program to find Factorial of a given number.					
19	Write an 8051 C program to find the Square of a number (1 to 10) using Look-Up Table.					
20	Write an 8051 C program to count the number of Ones and Zeros in two consecutive memory locations.					
21	Write an 8051 C Program to rotate stepper motor in Clock & Anti-Clockwise direction.					
22	Write an 8051 C program to Generate Sine & Square waveforms using DAC interface.					
	Average					

### 8051 Basic Component

- 4K bytes internal ROM
- 256 bytes internal RAM
- Four 8-bit I/O ports (PO P3).
- Two 16-bit timers/counters
- One serial interface



8051 Schematic Pin out





**Block Diagram** 



## **CIT GUBBI ECE DEPT**

**On-Chip Memory Internal RAM** 



Registers



### PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	• F0	RS1	RS0	ov	<u> </u>	Р			
CY	PSW.7	Carry Fla	ag.					-		
AC	PSW.6	Auxiliary	Carry Flag	5.						
<b>F</b> 0	PSW.5	Flag 0 av	vailable to th	ne user for g	eneral purp	oose.				
RS1	PSW.4	Register	Bank selecto	or bit 1 (SEI	E NOTE 1)	).				
RS0	PSW.3	Register	Bank selecto	or bit 0 (SEI	E NOTE I)	).				
ov	PSW.2	Overflow	Flag.							
_	<b>PSW.1</b>	User defi	nable flag.							
Р	<b>PSW</b> .0	Parity fla '1' bits in	g. Set/clear the accumi	ed by hardw ulator.	are each in	struction c	ycle to i	ndicate a	n odd/eve	n number of

#### NOTE:

1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH



Some 8-bit Registers of the 8051

Data types	Bits	Bytes	Value range
• Bit	1		0 to 1
• Signed char	8	1	-128 to +127
• Unsigned char	8	1	0 to 255
• enum	8\16	1\2	-128 to +127 or -32768 to +32767
<ul> <li>Signed short</li> </ul>	16	2	-32768 to +32767
• Unsigned short	16	2	0 to 65535
<ul> <li>Signed int</li> </ul>	16	2	-32768 to +32767
<ul> <li>Unsigned int</li> </ul>	16	2	0 to 65535
<ul> <li>Signed long</li> </ul>	32	4	-2147483648 to 2147483647
<ul> <li>Unsigned long</li> </ul>	32	4	0 to 4294967295
• Float	32	4	±1.175494E-38 to ±3.402823E+38
• sbit	1		0 to 1
• sfr	8	1	0 to 255

#### **Program execution Procedure**

Step1: Double click Keil uVision5 icon, go to project and select create new uVision project.



### Step3: Creating a new file.

Wision File Edit View Project Flash Debug Peripherals Tools S	VCS Window Help		– ¤ ×
日 2 2 3 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	8、存存///////////////////////////////////	- 🍳	
🖉 🕮 🖉 • 🖳 🞇 🔤 🖉 🖉 🖉	h = + - +		
Project 🛛 🖬	] Create New Project - → ← ↑ 🔒 → This PC → Desktop → EMD PGM → 💍 🔎 Search EMD PG	×	
	Organize - New folder	i∷ <b>- ?</b>	
	<ul> <li>This PC</li> <li>Dobjects</li> <li>Desktop</li> <li>Downloads</li> <li>Music</li> <li>Pictures</li> <li>Videos</li> <li>Local Disk (C;)</li> <li>New Volume (D; √</li> </ul>	Type	
^	Hide Folders Save	Cancel	
<u>■ P</u>			
Build Output			<b>₽</b> ×
			<u></u>
<			>
🕂 🔎 Search 🛛 🚱 🚾 🦷	ø		🗋 🕄 🎽 🎎 🔀 🗛 12:26 📮

Step4: Selecting device.

➢ Go to Microchip and select AT89C51 click ok.

CAUSers'Adm. Crestop/EMD BLWADD Borro- uVision for File Edit View Project Flass Debug Peripherals (15 Peripherals (15) Project (15) Project (15) Cause (15)	Commercialize Lice 1 BBB C X Webw + BBB A A C X Select Device for Target Target 1' X Device X Vendor qurknown> Device : qurknown> Toolet: qurknown>	DEPT
	Microchip         Image: Constraint of the second data of	
E P OB   OF   O		a 🖬
<		CAP' NUM SCRI OVR R /W
📒 🔎 Search 🛛 🚱 🔐 🚾 🤻	<u>ei</u>	📋 😍 🎽 🈫 🕀 d <sub>x</sub> 12:29 📮

### Step5: Select option NO.

😨 C-\Users\Admin\Desktop\EMD PGM\ADD_uvproj - µVision [Non-Commercial Use License]	– 🗇 🗙
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help	
□ 22 22 24 12 12 12 12 12 12 12 12 12 12 12 12 12	
🔗 🖸 🗃 🤪 - 😳 🙀 Target 1 🔤 🔊 🛔 着 🔶 🗇 🏟	
Project 4 🖬	
µVision × ② Copy 'STARTUP.AS1' to Project Folder and Add File to Project ? ▼ US NO ▼ US NO	
suna Output	4 🛛
	^
	~
	CAP NUM SCRL OVR R /W
🗄 🔎 Search 🔞 🔯 🚾 📧 🍻	🗓 😍 🎽 🤔 🕀 🗛 12:31 📮

Step6: Write a code by selecting empty document.

C:\Users\Adm Desktop\EMD	M\ADD proj - µVision fon-Commercianse Licent		
File Edit View Project Flas	Debug Peripherals Is Store We by F		
🛛 🖉 🎬 🥔 • 📖 🞇	Target 1 🖂 🚿 🚠 🖶 🗇 🛞		
Project 🛛 🗖 🔀	Text1*		▼ X
Project: ADD	<pre>#includecreg51.h&gt; void main () { unsigned int acc; acc=%t234; while(1); }</pre>		
נו אין פאר פאר דיין פאר נו דיין פאר נו דיין דיין פאר דיין דיין פאר דיין דיין פאר	¢		>
Build Output			÷ 🗵
			~
<			>
		Simulation	LIT C:2 CAP NUM SCRL OVR R/W
⊥			🗋 🤝 🏹 🏈 🤀 🖓 🖓 12:35 📮

Step7: Save empty document in the form of document name.C file. (Eg:add.C)

For 8051 uc save empty document in the form of document name.asm file. (Eg:add.asm)



Step8: Go to Target in that select source group1.Right click source group1 and add existing file to group source group 1.

🔣 C:\Users\Admin\Desktop\EMD PGM\ADD.uvproj - μVision [Non-Commercial Use License]		- 🛛 ×
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
□ 🐸 🖬 🖇   3 🗠   4 →   10 株 株 株   27 年 川 川 🐸 📃 🔜 4	•   🍭 -   • · · 🔗 🎪 -   🖬 -   🔦	
🧼 🏥 🎬 🧼 📲 🞇 🛛 Target 1 🛛 🖳 🔊 👘 🏙		
Project 4 🖬 🗋 add.c		▼ ×
Project: ADD     1 #include <reg51.h></reg51.h>		
E page Target 1 2 void main ()		
Source Original Source Group 1' Alt+F7		
Add New Item to Group 'Source Group 1'		
Add Existing Files to Group 'Source Group 1'		
Remove Group 'Source Group 1' and its Files		
Rebuild all target files		
Build Target F7		
📥 Manage Project Items		
Show Include File Dependencies		
		,
Build Output		4 🗵
		^
		~
<		>
Add Existing Files to current Project Group	Simulation	L:7 C:2 CAP NUM SCRL OVR R/W
🛨 🖓 Search 🛛 👹 🚾 🔏 🙍		[], 💱 🎽 🤔 🤀 🗛 12:39 📮

Step9: Selecting C file (Eg:Add.C) from source group1 add and close.

For 8051 uc Selecting asm file (Eg:Add.asm) from source group1 add and close.

😨 C:\Users\Admin\Desktop\EMD PGM\ADD.uvproj - µVision [Non-Commercial Use License]		— 🗇 🗙
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
File       Edit View Project Flash Debug Peripherals Tools SVCS Window Help         Image: Source Group 1       Image: Source Group 1         Image: Source Group 1 <td< th=""><th>Add Files to Group 'Source Group 1'</th><th>DEPT</th></td<>	Add Files to Group 'Source Group 1'	DEPT
	Varie     Date modified       Lisings     05-03-2024 12:32       Objects     05-03-2024 12:32       add.c     05-03-2024 12:38	
		,
suna Uutput		u 14
	Simulation	2 CAP NUM SCRU OVE RAW
🖽 🔎 Search 🛛 👰 🚾 🚾 👼	Jimaicon	🗇 😍 🎽 🥵 🖓 🔒 🗸

Step10: Save and Build the target file.

🔞 C:\Users\Admin\Desktop\EMD PGM\ADD.uvproj - μVision [Non-Commercial Use License]		– o ×
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
🗋 😂 📓 👗 🖄 💙 🍽 🖛 🔿 🥐 🗮 🕾 🦉 🦉 🦉 🎼 🎼	🖂 🗟 🍬 🎯 र 🛛 🔶 🔗 🚳 र 🗐 🕄	
🤣 🍱 🕮 🤣 - 🤐 🖾 Target 1 🔤 🔊 🔚 🖶 🚸 🐡 🏟		
<pre>Project Build (F7) Build target Ifes Build target Ifes Build target Ifes Bource Group 1 Build target Ifes Bource Group 1 Bource</pre>		• ×
▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲		>
Build Output		а <u>х</u>
Program Size: data=11.0 xdata=0 code=23 ".\Objects\ADD" - 0 Error(s), 0 Warning(s). Build Time Elapsed: 00:00:01		
<		>
Build target files	Simulation	L:7 C:2 CAP NUM SCRL OVR R/W
🕂 🔎 Search 🛛 🚱 🔐 🚾 🧏 🐠		📋 🔁 🎽 🎎 🤀 🗛 12:43 📮

Step11: Debug the program by selecting debug icon and click ok.

躍 C:\Users\Admin\Desktop\EMD PGM\ADD.uvproj - µVision [Non-Commercial Use License]		– 🛛 ×
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
Image: Transmission of the state of the	×	DEPT
<b>▲</b>		
□ <b>□ □ □ □ □ □ □ □ □ □</b>		>
Build Output		4 🗙
Program Size: data=ll.0 xddsta=0 code=23 ".\Objects\DDP" -0 Error(s), 0 Warning(s). Build Time Elapsed: 00:00:01		Ĵ
<		>
	Simulation	L:7 C:2 CAP NUM SCRL OVR R/W
🕂 🖓 Search 🛛 👹 🚾 🚾 🕫		📋 🛟 🎽 🎎 🤀 🕁 12:46 📮

Step12: Run the program by selecting run icon.

File Edit	Admin\Desktop\E View Project	MD PGN Flash	NADD.uvproj - µVision [Non-Commercial Use License] Debug Peripherals Tools SVCS Window Help					- 1	
	1 🖉   X 🖬 I	<b>2</b>   •7	(○) ← →   作良良良 (詳詳/////////////////////////////////	🛦 🌾 🔍 📲 🔶 🖉 🚓	- 🖻 - 🔧				
🏦 🖹	9 <b>7</b> 69 (P	*()	> 🖸 🕼 🖬 😂 🖉 • 🔲 • 📴 • 🖬 • 🕅 • 🛠 •						
Registers 👱	Run (F5)	ass	mbly						<b>д X</b>
Register	Start code execu	tion	2: void main ()						^
Regs			3: 1 4: unsigned int acc:						
r0	0x00		5: acc=0x1234:						~
r2	0x00	<							>
r3	0x00		add.c						<b>▼</b> ×
r4	0x00		1 #include <reg51.h></reg51.h>						
r6	0x00		2 void main () 3 (						
r7	0x00		4 unsigned int acc;						
a oys	0x00		5 acc=0x1234;						
Ь	0x00		<pre>6 while(1); 7 }</pre>						
sp sp	0x09								
dptr	0x00								
PC S	6 C:0x								
states	s 389 0.00								
	0×00								
Desired 2	- De sistere								
He Project	Registers	1	_						-
Command			4 🖬 Ca	all Stack + Locals					ф 🔀
Load "C:\	\Users\\Adm	ze Lin in\\De	LT: 2K sktop\\EMD PGM\\Objects\\ADD"	Name	Location/Value	Туре			
<			, E	MAIN	C:0x080C				<u> </u>
>			acc	0x0000	uint			-	
ASM ASSIGN BreakDisable BreakEnable BreakLill BreakList BreakAccess									
Start code execution Simulation t1: 0.00019450 sec				t1: 0.00019450 sec	L:5 C:1	CAP NUM SC	RL OVR R/W		
🗄 🔎 Search 🚯 🔯 📠 📧 👩				:47 📮					

#### Step13: To check the results stop program by selecting stop icon.

🔣 C:\Users\Admin\Desktop\EMD PGM\ADD.uvproj - μVision [Non-Commercial Use License]		- o ×
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
[[□] 😂 🖬 🗿   & ☜ 🎕   🤊 ୯   ← →   隆 陞 陞 数   建 建 /// /// [ 🖉 📃	🗟 🎢 🔍 - 🕒 🔗 🧟 - 🔲 - 🔦	
詩 💷 🐵 🕑 🕑 😕 🖕 🖸 👰 🖬 🚍 🖓 - 💷 - 📴 - 💷 - 👮 - 🙊 -		
Registers Stop mbly		<b>4 </b>
Project       Status       Status	I ECE DE	PŢ.
Command 📮 🖬	Call Stack + Locals	џ 🗵
Running with Code Size Limit: 2K	Name Location/Value Type	
Load "C://DEES/WAMLIN/DESKUD/LED FOR//DDJECES/WDD"	P         MAIN         C:0x080C             ∞         acc         0x0000         uint	
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess	Call Stack + Locals Memory 1	
Stop code execution	Simulation t1: 0.00019450 sec L:5 C:1	CAP NUM SCRL OVR R/W
🕂 🔎 Search 🛛 🚱 🚾 🚾 🛋	Ĉ 🕈 🎽	终 🕀 🔩 12:49 📮

Date:

### Write an ALP to move a block of 5data bytes from RAM locations 20H to 40H

ORG 00H MOV R2,#5 MOV R0,#20H MOV R1,#40H UP: MOV A,@R0 MOV @R1, A INC R0 INC R1 DJNZ R2, UP HERE: SJMP HERE END

**Result:** 

Before execution: Go to memory window-1 type I:0x0020 and enter 01,02,03,04,05

### **CIT GUBBI ECE DEPT**

After execution: go to memory window-2 type I:0x0040 and observe the data.

REPEAT:MOVXA,@DPTR MOV @R0, A **INCDPTR** INC R0

memory location

**ORG** 0000H

DJNZR2, REPEAT

MOVDPTR,#2000H MOV R0, #50H MOV R2, #5

MOVDPTR,#2050H MOV R0, #50H MOV R2, #5

REPEAT1:MOVA,@R0 MOVX @DPTR,A **INCDPTR** INC P0 DINZR2 REPEAT GUBBIECE DEPT END

**Result:** 

**Experiment No. 2** Write an ALP to transfer 5 bytes of data from source 2000h to destination 2050h using external

Microcontroller Lab (BECL456A)

**Page** 16

Date:

Date:

Write assembly language program to exchange 5 bytes of data at internal memory20H location to 40H location

ORG 0000H MOV R0,#20H MOV R1,#40H MOV R2, #5 UP:MOV A,@R0 XCH A, @R1 MOV @R0, AINC R0 INC R1 DJNZR2,UP HERE:SJMPHERE END

## Result: CIT GUBBI ECE DEPT

Date:

Write an ALP to exchange 5 bytes of data from external memory location 0000H with internal memory 10H location.

ORG 0000H MOVDPTR,#0000H MOV R0, #10H MOV R2, #5

REPEAT:MOVXA,@DPTR XCH A,@R0 MOVX@DPTR,A INC DPTR INCR0 DJNZ R2,REPEAT HERE:SJMPHERE END

2024-25

Date:

Write an ALP to add the byte in the RAM at 34h and 35h,store the result in the register R5 (LSB) and R6 (MSB), using Indirect Addressing Mode.

ORG 0000H MOVR0,#34H MOV A,@R0 INC R0 ADD A,@R0 MOV R5,A JNC DOWN MOV R6,#1 DOWN:NOP HERE:SJMPHERE END

Date:

Write an ALP to subtract the bytes in Internal RAM 34h& 35h store the result in register R5 (LSB) & R6 (MSB).

ORG 0000H MOV A,34H SUBBA,35H MOV R5,A JNC DOWN MOV R6,#1 DOWN:NOP HERE:SJMPHERE END

Date:

Write an ALP to multiply two 8-bit numbers stored at 30h and 31h and store16- bit result in 32h and 33h of Internal RAM.

Program ORG 0000H MOVA,30H MOVB,31H MUL AB MOV32H,A MOV33H,B HERE:SJMPHERE END

Date:

Write an ALP to perform division operation on 8-bit number by 8-bit number.

ORG 0000H MOVA, 30H MOVB, 31H DIV AB MOV32H, A MOV33H, B HERE: SJMP HERE END

Write an ALP to separate positive and negative in a given array

**ORG** 0000H MOVDPTR,#1000H MOV R0,#30H MOVR1.#3AH MOV R2,#05 UP:MOVXA,@DPTR INC DPTR **RLCA** JBPSW.7,NEXT RRC A MOV@R0,A INC **R0SJMPDO** WN NEXT:RRCA MOV @R1,A INC R1 DOWN:DJNZR2,UP HERE: SJMP HERE END CIT GUBBI ECE DEPT

Date:

Write an ALP to separate even or odd elements in a given array.

**ORG** 0000H MOV DPTR,#1000H MOV R0,#30H MOVR1,#3AH MOV R2,#05 UP:MOVXA,@DPTR INC DPTR RRC A JBPSW.7,NEXT RLC A MOV@R0,A INCR0 SJMP DOWN NEXT:RLCA MOV @R1,A INC R1 DOWN:DJNZ R2,UP HERE: SJMP HERE END

## **CIT GUBBI ECE DEPT**

**Date:** 

Page 24

Write an ALP to arrange the numbers in Ascending & Descending order.

**ORG** 0000H MOVR2,#04H OUTER:MOVR0,#40H MOV A.R2 MOV R1,A INNER:MOVA,@R0 INC R0 MOV B,@R0 CJNE A, B, NEXT NEXT: JCDOWN MOV @R0,A DEC R0 MOV@R0,B INC R0 DOWN:DJNZR1,INNER DJNZ R2, OUTER END

## **CIT GUBBI ECE DEPT**

Date:

Microcontroller Lab (BECL456A)

Write an ALP to find Largest & Smallest number from a given array starting from 20h & store it in Internal Memory location 40h.

ORG 00H MOVR3,#04H MOVR0,#20H MOVR1,#40H MOV A,@R0 INC R0 UP:MOVB,@R0 CJNEA,B,NEXT NEXT:JNC NOCARRY MOV A,@R0 NOCARRY: INC R0 DJNZ R3,UP MOV@R1,A END

## **CIT GUBBI ECE DEPT**

Date:

### **COUNTER OPERATION PROGRAMS**

#### **Experiment No. 13**

Date:

Write an ALP for Decimal UP-Counter

ORG 0000H MOVA,#00H UP:ACALL DELAY ADD A,#01H DA A JNZUP HERE: SJMP HERE

DELAY:MOVR1,#0FFH DECR1:MOV R2,#0FFH DECR: MOV R3,#0FFH HERE1:DJNZR3,HERE1 DJNZ R2,DECR DJNZR1,DECR1 RET END

Date:

#### Write an ALP for Decimal DOWN-Counter

ORG 0000H MOVA,#99H UP:ACALLDELAY ADD A,#99H DA A JNZUP HERE: SJMP HERE

DELAY:MOVR1,#0FFH

DECR1:MOV R2,#0FFH DECR: MOV R3,#0FFH HERE1:DJNZR3,HERE1 DJNZ R2,DECR DJNZR1,DECR1 RET END

Write an ALP for Hexadecimal UP-Counter.

ORG 0000H MOVA,#00h BACK:ACALLDELAY INC A JNZ BACK HERE: SJMP HERE

DELAY:MOVR1,#0FFH DECR1:MOV R2,#0FFH DECR: MOV R3,#0FFH HERE1:DJNZR3,HERE1 DJNZ R2,DECR DJNZR1,DECR1 RET END

## **CIT GUBBI ECE DEPT**

2024-25

Date:

Date:

Write an ALP for Hexadecimal DOWN-Counter.

ORG 0000H MOVA,#0FFh BACK:ACALLDELAY DEC A JNZBACK HERE: SJMP HERE

DELAY:MOVR1,#0FFH DECR1:MOV R2,#0FFH DECR: MOV R3,#0FFH HERE1:DJNZR3,HERE1 DJNZ R2,DECR DJNZR1,DECR1 RET END

### PART-II

### **C PROGRAMMING**

### **Experiment No. 17**

```
Write an 8051 C program to find the sum of first 10 integer numbers.
#include <reg51.h>
void main ()
{
    unsigned char i;
    unsigned int sum=0;
    for(i=1;i<=10;i++)
    {
    sum=sum+i;
    }
    P0=sum;
    while(1);
  }</pre>
```

Date:

Re Be	ore election	<b>UBBIECE DEPT</b>
1	sum.	0x00
tches		
Na.	Locals ∧ Watch #1 ∧	Watch #2 \ Call Stack /

### After execution



### Write an 8051 C program to find factorial of a given number.

```
#include <reg51.h>
void main()
{
    unsigned int num=5,fact=1;
    while(num>0)
    {
    fact=fact*num;
    num--;
    }
    P1=fact;
    while(1);
}
```

### Result

**Before execution** 

×	Name	Value			
	num	0x0000			
		MDDIECE NEDT			
ches					
Wate	Locals Watch #1	Watch #2 \ Call Stack /			
Af	After execution				

X	Name	Value			
1	num	0x0000			
	fact	0x0078			
Ë					
Wato	K ← ▶ ► Locals ( Watch #1 )	Watch #2 \ Call Stack /			
		Simulation	t1: 26.59210450 sec L:12 C:1		-
		,	, ,	, , , , ,	

Date:

Date:

### Write an 8051C program to find the square of a number (1to10) using look-up table.

```
#include <reg51.h>
void main ()
{
    unsigned int result, num=4;
    unsigned int tab[]={0x0,0x1,0x4,0x9,0x16,0x25,0x36,0x49,0x64,0x81,0x100};
    if(num<10)
    result =tab[num];
    else
    result=0;
    while(1);
  }</pre>
```

### Result:

#### **Before execution**

×	Name	Value
1	result	Gx0000
	num	0x0000
	Ė tab	D:0x0C [ ]
	<b>CIT Gl</b>	BBIECE DEPT
	[4]	0x0000
	[5]	0x0000
	[6]	0x0000
	[7]	0x0000
	[8]	0x0000
	[9]	0x0000
Ę.	[10]	0x0000
Wat	Locals / Watch #1 /	Watch #2 \ Call Stack /
		Simulation t1: 0.00019450 sec L:4 C:1

### After execution

×	Namo	Value
	Name	
	result	UKUU IIS
	num	0×0004
	Etab	D:0x0C [ ]
	[0]	0x0000
	[1]	0x0001
	[2]	0x0004
	[3]	0x0009
	[4]	0x0016
	[5]	0x0025
	[6]	0x0036
	[7]	0x0049
	[8]	0x0064
	[9]	0x0081
ŝ	······ [10]	0x0100
Wat	Id I ▶ ► Locals / Watch #1 /	Watch #2 \ Call Stack /
		Simulation t1: 24.95081550 sec L:10 C:1

Experiment No. 20Date:Write an 8051 C program to count the number of ones and zeros in two consecutive<br/>memory locations.

```
#include <reg51.h>
void main ()
{
unsigned char a[]={0xfa,0xfa},i;
unsigned char ones, zeros;
CY=0;
for(i=0;i<8;i++)
{
a[0]>>=1;
if(CY==1) ones++;
else zeros++;
a[1]>>=1;
if(CY==1) ones++;
else zeros++;
}
while(1);
}
                                                                                                            GUBBI ECE DEPT
 Re
                    ult
 Before execution
                                                                                                                                         Value
D:0x08
                 Name
                                                                                                                                           0x00
0x00
                                   ···· [0]
···· [1]
                                                                                                                                            0x00
                                                                                                                                           0x00
0x00
                            ones
                            zeros

    Image: State of the state o
                                                                                                                                                                    Simulation
                                                                                                                                                                                                                                                                   t1: 0.00019450 sec L:4 C:1
                                                                                                                                                                                                                                                                                                                                                                                              CAP NUM
After execution
```

#### Value D:0x08 [ Nar а [0] [1] $0 \times 00$ 0x00 i. 0x08 0x0C ones 0x04 zeros K K Locals (Watch #1 ) Watch #2 ) Call Stack / t1: 20.26474400 sec L:16 C:1 CAP NUM Simulation

### PART-III

### HARDWARE INTERFACE PROGRAMS

**Experiment No. 21** 

#### Write an 8051C program to generate the square, using DAQ.

#include <reg51.h>
void delay(unsigned int x)
{
for(;x>0;x--);
}
main()
{
unsigned char on=0xff,off=0x00;
P0=on;
delay(100);
P0=off;
delay(100);
}
CITGUBBBIECEDEPT

Result: After execution



Write an 8051C program to rotate motor in clock and anti-clockwise direction

Date:

### SINE WAVE:

#include<reg51.h>
main()
{
Unsigned char
a[]={128,192,238,255,238,192,128,64,17,0,
17, 64};
int i;
while(1)
{
for(i=0;i<12;i++)
P0=a[i];
}}</pre>

### Date:

### Write an 8051C program to rotate motor in clock and anti-clockwise direction

CLOCKWISEROTATION	CLOCKWISEROTATION	ANTICOLCKWISE
#include <reg51.h></reg51.h>	(ANOTHER TYPE)	ROTATION
Void delay(unsigned int x)		#include <reg51.h></reg51.h>
{	#include <reg51.h></reg51.h>	Void delay(unsigned int x)
for(;x>0;x);	Void delay(unsigned int x)	{
}	{	for(;x>0;x);
main()	for(;x>0;x);	}
{	}	main()
Unsigned char val,i;	main()	{
while(1)	{	Unsigned char val ,i,j,k;
{	Unsigned char val;	while(1)
val=0x88;	while(1)	{
for(i=0;i<4;i++)	{	for(i=0;i<0x19;i++)
{	val=0x03;	{
P2=val;	P2=val;	val=0x88;
val=val>>1;	delay(5750);	for(j=0;j<4;j++)
		DEDT
		P2 N 1,
	deray(5750);	val=val>>1,
	val=0x0c;	delay(1000);
	P2=val;	<pre>}}</pre>
	delay(5/50);	tor(k=0;k<0x32;k++)
	val=0x06;	{
	P2=val;	val=0x11;
	delay(5750);	for(j=0;j<4;j++)
	}}	
		$P_{2}=val;$
		val = val < 1;
		]]]]