SCHEME : K

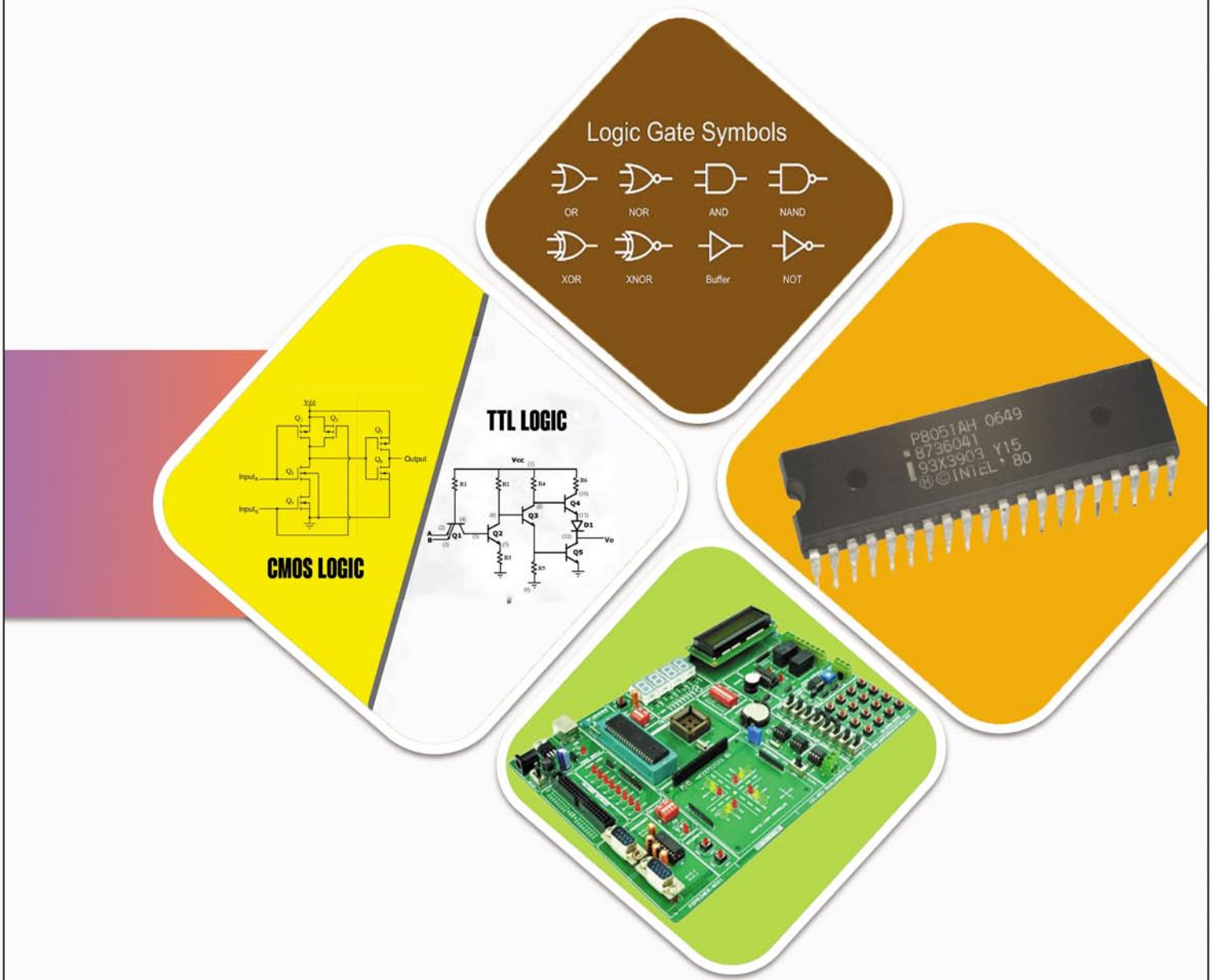
Name : _____

Roll No. : _____ Year : 20___ 20___

Exam Seat No. : _____

LABORATORY MANUAL FOR

DIGITAL ELECTRONICS AND MICROCONTROLLER APPLICATIONS (314324)



ELECTRONICS ENGINEERING GROUP

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI
(Autonomous) (ISO 9001: 2015) (ISO/IEC 27001:2013)

VISION

To ensure that the Diploma Level Technical Education constantly matches the latest requirements of technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

MISSION

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the changing technological and environmental challenges.

QUALITY POLICY

We, at MSBTE, are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation evaluation and monitoring system along with adequate faculty development Programs.

CORE VALUES

MSBTE believes in the followings:

- Education industry produces live products.
- Market requirements do not wait for curriculum changes.
- Question paper is the reflector of academic standards of educational organization
- Well-designed curriculum needs effective implementation too.
- Competency based curriculum is the backbone of need based programs.
- Technical skills do need support for life skills.
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.

A Laboratory Manual for

Digital Electronics And Microcontroller Applications

(314324)

Semester-IV

(EE/ EP)

Maharashtra State

Board of Technical Education, Mumbai

(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)

Maharashtra State
Board of Technical Education, Mumbai

(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)

4th Floor, Government Polytechnic Building, 49, Kherwadi, Bandra (East), Mumbai - 400051.

MAHARASHTRA STATE
BOARD OF TECHNICAL EDUCATION
Certificate

This is to certify that Mr. / Ms. …….. Roll No.……..… of Fourth Semester of Diploma in ……. of Institute……..……..…….. (Code: ……) has attained pre-defined practical outcomes (PROs) satisfactorily in course **Digital Electronics And Microcontroller Applications (314324)** for the academic year 20…..... to 20..…. as prescribed in the curriculum.

Place: …… Enrollment No: …….

Date: …… Exam. Seat No: ……

Course Teacher **Head of the Department** **Principal**

Seal of the Institute

Preface

The primary focus of any engineering laboratory/field work in the technical education system is to develop the much needed industry relevant competencies and skills. With this in view, MSBTE embarked on this innovative 'K' Scheme curricula for engineering diploma programmes with outcome- based education as the focus and accordingly, a relatively large amount of time is allotted for the practical work. This displays the great importance of laboratory work, making each teacher, instructor and student realize that every minute of the laboratory time needs to be effectively utilized to develop these outcomes, rather than doing other mundane activities. Therefore, for the successful implementation of this outcome-based curriculum, every practical has been designed to serve as a ***'vehicle'*** to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'K' scheme laboratory manual development team designed the practicals to ***focus*** on the ***outcomes***, rather than the traditional age old practice of conducting practicals to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the predetermined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read through the concerned practical procedure that they will do the next day and understand the minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. The students will then become aware about the skills they will achieve through the procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

The electrical diploma holder has to work in industry as a technical person in middle level management. He/she has to work as production, maintenance, testing engineer in various domain like power generation, transmission, distribution, traction etc. and has to deal with different electrical measurements. He/she also has to deal with advanced, automated and sophisticated equipment that are used in modern techniques. While performing the above task he/she has to measure different electrical and electronic parameters with testing, therefore he/she must require the skills for these measurements and a broad idea of different meters and equipment. Equipment may contain digital and microcontroller based embedded systems, and this course is intended to develop the skills to maintain and solve the application problems related to measurement, control and automation based on microcontrollers.

Although best possible care has been taken to check for errors (if any) in this laboratory manual, perfection may elude us as this is the first edition of this manual. Any errors and suggestions for improvement are solicited and highly welcome.

Programme Outcomes (POs) and Program Specific Outcomes (PSOs) to be achieved through Practicals of this Course

Following programme outcomes are expected to be achieved through the practical of the course.

PO1: Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the broad based Electrical engineering problems.

PO2: Problem analysis: Identify and analyze well-defined Electrical engineering problems using codified standard methods.

PO3: Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of Electrical systems components or processes to meet specified needs.

PO4: Engineering Tools, Experimentation and Testing: Apply modern Electrical engineering tools and appropriate technique to conduct standard tests and measurements.

PO5: Engineering practices for society, sustainability and environment: Apply appropriate Electrical technology in context of society, sustainability, environment and ethical practices.

PO6: Project Management: Use Electrical engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well- defined engineering activities.

PO7: Life-long learning: Ability to analyze individual needs and engage in updating in the context of Electrical technological changes.

List of Industry Relevant Skills

The following industry relevant skills of the competency 'Use digital electronics and microcontroller based systems' are expected to be developed in the students by undertaking the laboratory work in this practical manual.

a. Realize logic circuits using Boolean expressions.

b. Build simple combinational and sequential circuits.

c. Analyse the architecture of microcontroller ICs.

d. Write programs in assembly language for microcontrollers.

e. Interface the memory and various I/O devices to microcontrollers.

Practical- Course Outcome matrix

Course Outcomes (COs)

CO1- Apply knowledge of number system and logic circuits in working of digital system.
CO2- Build simple combinational and sequential circuits.
CO3- Access various registers in 8051 microcontroller.
CO4- Develop and execute programs in assembly language for microcontroller.
CO5- Use microcontroller in various applications.

Sr. No.	Title of the Practical	CO 1	CO 2	CO 3	CO 4	CO 5
1.	Verification of the truth table of AND, OR, NOT gates using ICs.	✓	-	-	-	-
2.	Building of AND, OR, NOT gates using NAND gate.	✓	-	-	-	-
3.	Building of AND, OR, NOT gates using NOR gate.	✓	-	-	-	-
4.	Building of Half adder and Half subtractor using Boolean expressions.	✓	-	-	-	-
5.	Building of Full adder and full subtractor using Boolean expressions.	✓	-	-	-	-
6.	Verification of operation of Multiplexer IC74151	-	✓	-	-	-
7.	Verification of operation of Demultiplexer IC74155	-	✓	-	-	-
8.	Testing the function of RS flip flop using NAND Gate.	-	✓	-	-	-
9.	Testing the function of JK flip flop using 7476.	-	✓	-	-	-
10.	Construction and testing of the functionality of D flip flop using IC 7476.	-	✓	-	-	-
11.	Construction and testing the functionality of T flip flop using IC 7476.	-	✓	-	-	-
12.	Implementation of a 4-bit ripple counter using7476.	-	✓	-	-	-
13.	Assembly language program (ALP) to perform addition of 8-bit data using various addressing modes.	-	-	✓	✓	-
14.	Assembly language program (ALP) to perform subtraction of 8-bit data using various addressing modes	-	-	✓	✓	-
15.	Assembly language program (ALP) to perform multiplication of 8-bit data, take the input data from port1 and display the output data on port 2	-	-	✓	✓	-

Sr. No.	Title of the Practical	CO 1	CO 2	CO 3	CO 4	CO 5
16.	Assembly language program (ALP) to perform division of 8-bit data, take the input data from port 2 & display the output data on port 0	-	-	✓	✓	-
17.	Assembly language program to transfer data from source to destination location of internal data memory.	-	-	✓	✓	-
18.	Assembly language program to transfer data from source to destination location of external data memory.	-	-	✓	✓	-
19.	Assembly language program to exchange data from source to destination memory location.	-	-	✓	✓	-
20.	Assembly language program to find the smallest number from the given data bytes stored in internal / external data memory locations.	-	-	✓	✓	-
21.	Assembly language program to find the largest number from the given data bytes stored in internal / external data memory locations.	-	-	✓	✓	-
22.	Assembly language program for arranging numbers in ascending order stored in external memory locations.	-	-	✓	✓	-
23.	Assembly language program for arranging numbers in descending order stored in external memory locations.	-	-	✓	✓	-
24.	Assembly language program MASK and SET particular bit of given register using 1) bit addressable instructions 2) Logical instructions.	-	-	✓	✓	-
25.	Assembly language program to get a rolling display on port 2.	-	-	✓	✓	-
26.	Interfacing of LED and switch with 8051 to turn ON / OFF the LED.	-	-	-	-	✓
27.	Interfacing of RELAY with 8051 to turn ON /OFF the LED.	-	-	-	-	✓
28.	Interfacing of 7-segment display with 8051 to give output as decimal number from 0 to 9.	-	-	-	-	✓
29.	Interfacing of LCD with 8051 microcontrollers to display the alphabets and decimal numbers.	-	-	-	-	✓
30.	Interfacing of stepper motor with 8051 microcontroller and write ALP to rotate stepper motor in clockwise and anti-clockwise direction at given angles.	-	-	-	-	✓

Guidelines to Teachers

1. Teacher should provide the guideline with demonstration of practical to the students with all features.
2. Teacher shall explain prior concepts to the students before starting of each practical.
3. Involve students in the performance of each experiment.
4. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
5. Teachers should give opportunities to students for hands-on experience after the demonstration.
6. Teacher is expected to share the skills and competencies to be developed in the students.
7. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected of the students by the industry.
8. Finally give practical assignments and assess the performance of students based on tasks assigned to check whether it is as per the instructions.
9. Teacher is expected to refer complete curriculum document and follow guidelines for implementation
10. At the beginning of the practical, which is based on the simulation, teacher should make the students acquainted with any simulation software environment.

Instructions for Students

1. Listen carefully to the lecture given by the teacher about course, curriculum, learning structure, skills to be developed.
2. Organize the work in the group and make a record of all observations.
3. Do the calculations and plot the graph wherever it is required in the practical
4. Students shall develop maintenance skills as expected by industries.
5. Student shall attempt to develop related hand-on skills and gain confidence.
6. Student shall develop the habits of evolving more ideas, innovations, skills etc. those included in scope of manual
7. Student should develop the habit to submit the practical on date and time.
8. Student should prepare well while submitting a write-up of exercise.

Content Page

List of Practicals and Progressive Assessment Sheet

Sr. No.	Title of the practical	Page No.	Date of performance	Date of submission	Assessment marks (25)	Dated sign. of teacher	Remarks (if any)
15.*	Assembly language program (ALP) to perform multiplication of 8-bit data, take the input data from port1 and display the output data on port 2	125					
16.*	Assembly language program (ALP) to perform division of 8-bit data, take the input data from port 2 & display the output data on port 0	132					
17.*	Assembly language program to transfer data from source to destination location of internal data memory.	139					
18.	Assembly language program to transfer data from source to destination location of external data memory.	147					
19.*	Assembly language program to exchange data from source to destination memory location.	153					
20.	Assembly language program to find the smallest number from the given data bytes stored in internal / external data memory locations.	160					
21.*	Assembly language program to find the largest number from the given data bytes stored in internal / external data memory locations.	168					
22.*	Assembly language program for arranging numbers in ascending order stored in external memory locations.	175					
23.	Assembly language program for arranging numbers in descending order stored in external memory locations.	181					
24.*	Assembly language program MASK and SET particular bit of given register using 1) bit addressable instructions 2) Logical instructions.	188					

Sr. No.	Title of the practical	Page No.	Date of performance	Date of submission	Assessment marks (25)	Dated sign. of teacher	Remarks (if any)
25.*	Assembly language program to get a rolling display on port 2.	194					
26.*	Interfacing of LED and switch with 8051 to turn ON / OFF the LED.	201					
27.*	Interfacing of RELAY with 8051 to turn ON /OFF the LED.	209					
28.	Interfacing of 7-segment display with 8051 to give output as decimal number from 0 to 9.	216					
29.	Interfacing of LCD with 8051 microcontroller to display the alphabets and decimal numbers.	224					
30.	Interfacing of stepper motor with 8051 microcontroller and write ALP to rotate stepper motor in clockwise and anti-clockwise direction at given angles.	234					
Total							

Practical No.1: Verification of the truth table of AND, OR, NOT gates using ICs.

I Practical Significance

Logic gates are the basic building block of all types of digital systems. There are few basic logical operations performed a number of times in digital systems like computers or control systems. The basic operations which are performed are AND, OR and NOT. Knowledge of functions of logic gates will help the students to build the digital circuits.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Apply knowledge of number system and logic circuits in working of digital system.

IV Laboratory Learning Outcome

Build AND, OR, NOT gates to verify its truth table.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

A logic gate is an electronic circuit, which makes logical decisions. It has only one output and one or many inputs. The output signal appears for certain combinations of input signals. Logic gates are the basic building blocks of all digital systems. These gates are AND, OR, NOT, NAND, NOR, EX-OR and EX-NOR gates. The basic operations are described below with the help of truth tables. In digital logic design, only two voltage levels or states are applied as input, and these states are generally referred to as Logic "I" and Logic "0", High and Low, or True and False. These two states are represented in truth tables as binary digits "1" and "0" respectively.

In digital circuits, a high impedance (hi-Z) also known as tri-state or floating output condition, is not being driven to any defined logic level by the output circuit. The signal is neither driven to a logical high nor low level; this third condition leads to the description “tri-stated”.

To verify the truth table of the gate, all possible input conditions are provided to the gate as per the number of inputs of the gate. For example, NOT gate has only one input so possible conditions are two (i.e. 0 and 1), because $2^n=P$, where n is number of inputs and P is number of combinations of input conditions, which defines number of rows in the truth table of the gate. So the truth table of NOT gate contains two rows and two columns, one column to write input ‘A’ condition and second column is for writing output for that particular input condition and rows for combinations of inputs. Similarly for 2-input AND or OR gate number of combinations are $2^2 = 4$.

Classification of Logic gates:

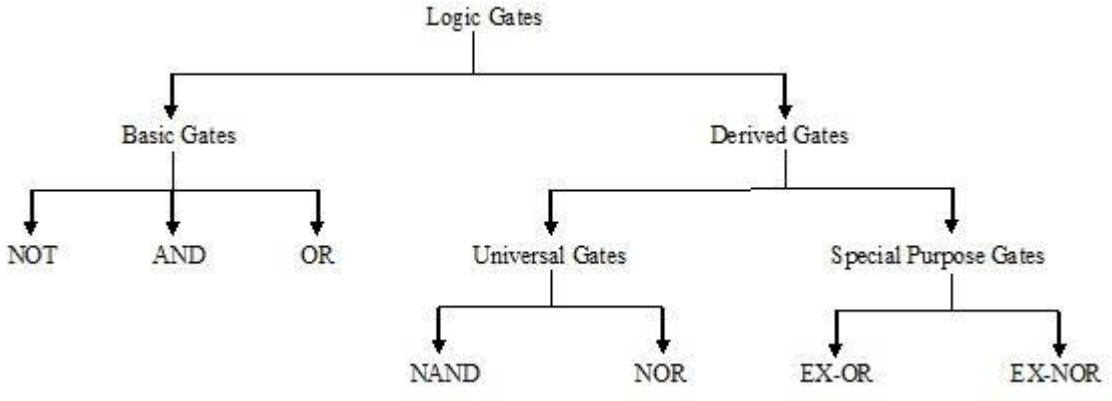


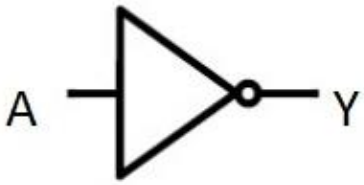
Figure 1.1 Classification of Logic gates

NOT Gate: NOT gate is also known as **Inverter**. It has one input A and one output Y. Output Y is a complement of input A.

$$Y = \text{NOT } A$$

$$Y = \overline{A}$$

Symbol



Truth Table

Input A	Output Y
0	1
1	0

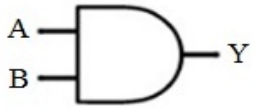
Figure 1.2 Symbol and Truth table of NOT gate

AND Gate: In the AND gate, the output of an AND gate is Logic 1(High) if and only if all the inputs are Logic 1(High). It has n inputs (n >= 2) and one output.

$$Y = A \textbf{ AND } B$$

$$Y = A \cdot B$$

Symbol



Truth Table

Input A	Input B	Output Y
0	0	0
0	1	0
1	0	0
1	1	1

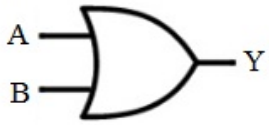
Figure 1.3 Symbol and Truth table of AND gate

OR Gate: In the OR gate, the output of an OR gate is Logic 1(High) if any one input is Logic 1(High) from the number of inputs. It has n inputs (n >= 2) and one output.

$$Y = A \textbf{ OR } B$$

$$Y = A + B$$

Symbol



Truth Table

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1.4 Symbol Truth table of OR gate

ICs used to perform basic gate operations-

Logic gate ICs have a wide range of applications in various fields. These ICs are used in digital electronics, communications systems, control systems, and computer systems. They are essential components in modern electronic systems and are constantly evolving to meet the demands of new applications. According to the function, some of the ICs from TTL logic family are as follows,

List of Logic Gate ICs and their functions

IC Number	**Function of the IC**
IC 7404	Hex Inverter (NOT Gate IC)
IC 7408	Quad 2-input AND Gate IC
IC 7432	Quad 2-input OR Gate IC
IC 7400	Quad 2-input NAND Gate IC
IC 7402	Quad 2-input NOR Gate IC

74XX Series of Logic Family:

This is the most widely used logic family. Following example shows how part numbering is used to identify the IC.

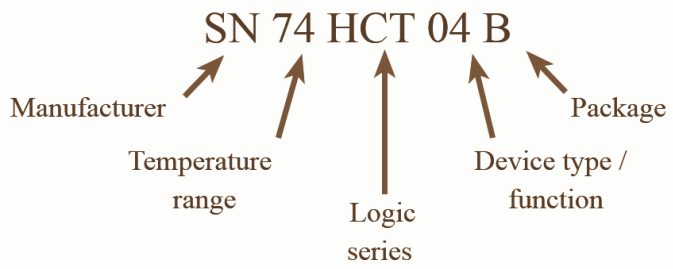


Figure 1.5 Part numbering scheme for 74XX00 series Logic ICs

The part numbers for 7400-series logic devices often use the following designators:

Manufacturer - This code normally consists of two letters and is a code normally used by a given manufacturer. SN is one used by Texas Instruments. Other manufacturers have their own codes that they place here.

Temperature range - This is indicated by these two figures. 74 indicates 0°C to 70°C commercial and 54 military: -55°C to +125°C. For most applications the 74XX series is perfectly acceptable and this series will be found in consumer devices.

Logic series - This is the sub-family. 7400 for example is the basic series, but there are many others.
Device - This indicates the device function / type. For example devices with 04 are hex inverters, etc. They are common across all sub-families.
Package code - This is the package suffix. It is necessary to refer to the manufacturer's data sheets as these codes vary between manufacturers.

For example-

Figure 1.6 Example of Logic Gate IC with part numbering

The specifications and features of 74LS08 (LS- Low Power Schottky) AND gate IC are as follows,

- The operating speed of IC 74LS08 is fast.
- This IC has less power consumption.
- It has Quadruple 2-Input AND Gate features.
- It is available in several packages like SOP, PDIP and SOIC.
- It is very simple to operate.
- It is much cheaper.
- The output of this IC is TTL so that it can operate with microcontrollers and devices.
- A single power supply can activate all logic gates within the IC.
- An AND gate is used without affecting other gates.
- The maximum current provided by this IC is 8mA.
- The operating temperature of this IC ranges from 4.75V to 5.25V.
- The recommended voltage of this IC must be 5V however this IC holds up to 7V max.
- The operating temperature ranges from 0 – 70 degrees however it has the capability to store -65 to 150 degrees temperature.

40XX Series of Logic family-

The 4000 series was the first family or series of CMOS ICs.

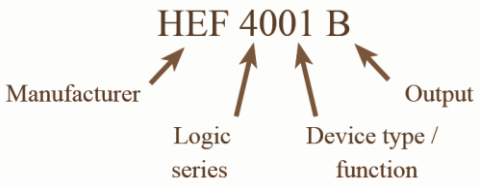
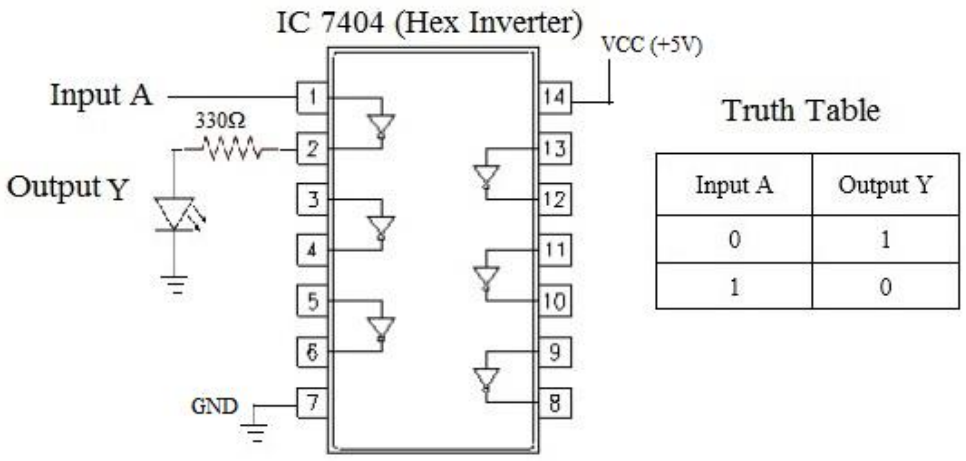


Figure 1.7 Part numbering for 4000 Series Logic families

VII Actual Circuit diagram used in laboratory with related equipment rating-

A) Suggestive Circuit diagram-

NOT Operation-

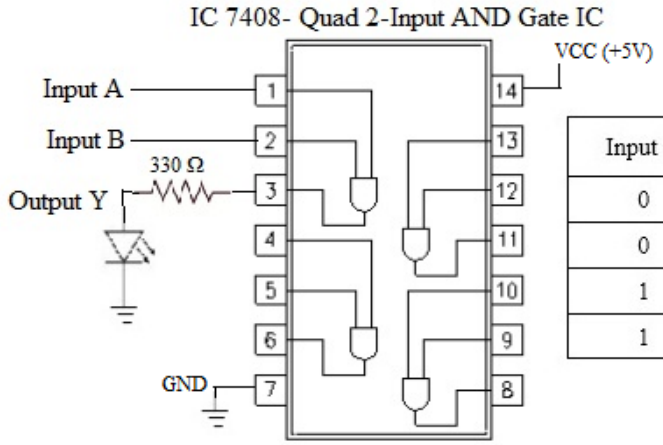


Truth Table

Input A	Output Y
0	1
1	0

Figure 1.8 Circuit diagram to verify truth table of NOT operation using IC 7404.

AND Operation-

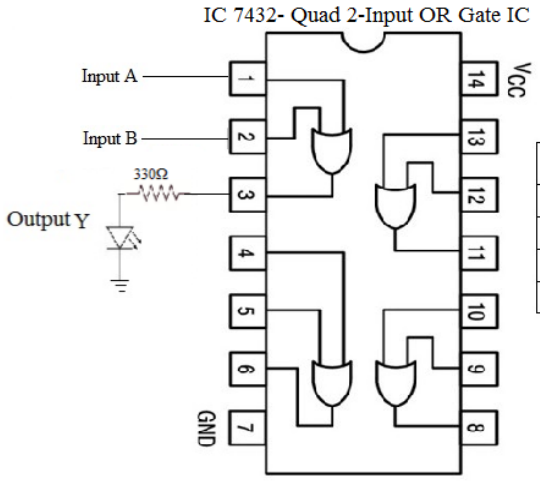


Truth Table

Input A	Input B	Output Y
0	0	0
0	1	0
1	0	0
1	1	1

Figure 1.9 Circuit diagram to verify truth table of AND operation using IC 7408.

OR Operation-



Truth Table

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1.10 Circuit diagram to verify truth table of OR operation using IC 7432

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5cm X 17 cm.	1
5	IC	7404, 7408, 7432 or any IC of equivalent function..	1 each
6	LED	Red/Yellow color 5 mm.	1
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	1

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition, then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of all ICs used.
5. Make the connections as per the logic diagram given in Figure 1.8.
6. Apply logic inputs and verify the truth table for each gate by observing LED status for output.
7. Measure the output voltage with the help of a digital multimeter.
8. Write the reading in each observation table.
9. Repeat the process for all logic gates by making the connections as shown in Figure 1.9 and 1.10.

XI

XII Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XIII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIV Observation Table

Table 1: Verification of the truth table of NOT gate using IC.

Sr. No.	**Input**	**IC 7404 - Output Y**	
	A	LED Status (ON/Off)	Output Voltage (V)
1.	0		
2.	1		

Table 2: Verification of the truth table of AND, OR gates using ICs.

Sr. No.	**Inputs**		**IC 7408 - Output Y**		**IC 7432 - Output Y**	
	A	B	LED Status (ON/Off)	Output Voltage (V)	LED Status (ON/Off)	Output Voltage (V)
1.	0	0				
2.	0	1				
3.	1	0				
4.	1	1				

XV Result(s)

……

……..

……..

XVI Interpretation of results

……..

……..

……..

……..

XVII Conclusion and recommendation

……..

……..

……..

……

XVIII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. State the logic low and logic High levels of TTL ICs and CMOS ICs?
2. Write the name, specification, cost and manufacturer of the major electrical equipment used for the experiment.
3. Draw logic level diagrams for positive logic systems and negative logic systems used in digital design.

[Space for Answers]

XIX References/Suggestions for further reading

1. https://de-iitr.vlabs.ac.in/exp/truth-table-gates/theory.html
2. https://drive.google.com/file/d/13SkA6bhfxOfc204nhKyIrsHYUjkLbote/view
3. https://drive.google.com/file/d/1iHUjR5wvsRhNJ2hs7TESiyrLZyD-ZmGS/view?amp;ts=5f02cb50%3D%3D
4. https://www.ovaga.com/blog/transistor/7404-integrated-circuit-ic-datasheet-pinout-pin-diagram-truth-table
5. https://www.ovaga.com/blog/transistor/7408-integrated-circuit-datasheet-pinout-pin-diagram-truth-table
6. https://www.futurlec.com/IC74LS00Series.shtml
7. https://circuitdigest.com/electronic-circuits/or-gate-circuit
8. https://www.alldatasheet.com/datasheet-pdf/pdf/50893/FAIRCHILD/7408.html

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.2: Building of AND, OR, NOT gates using NAND gate.

I Practical Significance

NAND gate is actually a combination of two logic gates: AND gate followed by NOT gate. So its output is a complement of the output of an AND gate. This gate can have a minimum of two inputs; output is always one. By using only NAND gates, it is possible to realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NOR. So this gate is also called universal gate

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Apply knowledge of number system and logic circuits in working of digital system.

IV Laboratory Learning Outcome

Build AND, OR, NOT gates using NAND gate and verify its truth table of NAND gate as universal gate.

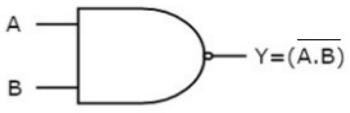
V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

NAND and NOR gates are universal gates because any Boolean expression can be realized using these gates. The NAND gate has the ability to perform three operations such as AND, OR and NOT. This gate is a combination of NOT and AND gates. The NAND gate output is equal to the inverse of the AND gate. The symbol and truth table is as shown below.

Logic Symbol



Truth Table

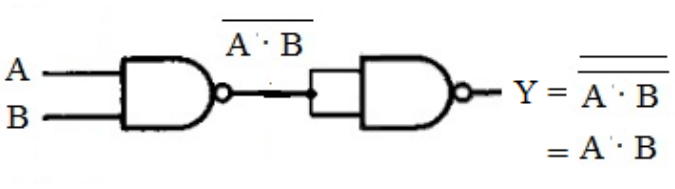
Input A	Input B	Output Y
0	0	1
0	1	1
1	0	1
1	1	0

Figure 2.1 Logic symbol and truth table of NAND gate

NAND gate as AND gate:

A NAND produces a complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.

AND Gate using NAND Gates



Truth Table of AND Gate

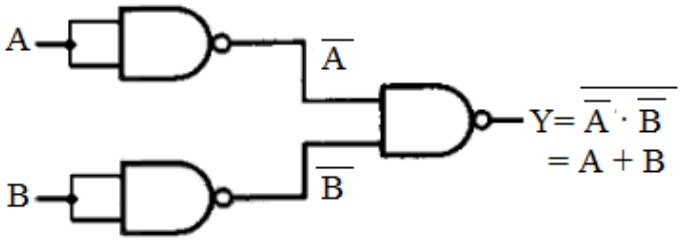
Input A	Input B	Output Y
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2.2 Connection diagram for NAND gate as AND gate.

NAND gate as OR gate:

From De-Morgan's theorems, a bubbled NAND gate is equivalent to the OR gate.

OR Gate using NAND Gates



Truth Table of OR Gate

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 2.3 Connection diagram for NAND gate as OR gate.

NAND gate as NOT gate: A NOT produces complement of the input. It can have only one input, tie the inputs of a NAND gate together. Now it will work as a NOT gate.

NOT Gate using NAND Gate

Truth Table of NOT Gate

Input A	Output Y
0	1
1	0

Figure 2.4 Connection diagram for NAND gate as NOT gate.

VII Actual Circuit diagram used in laboratory with related equipment rating-

A) Suggestive Circuit diagram-

AND Gate:

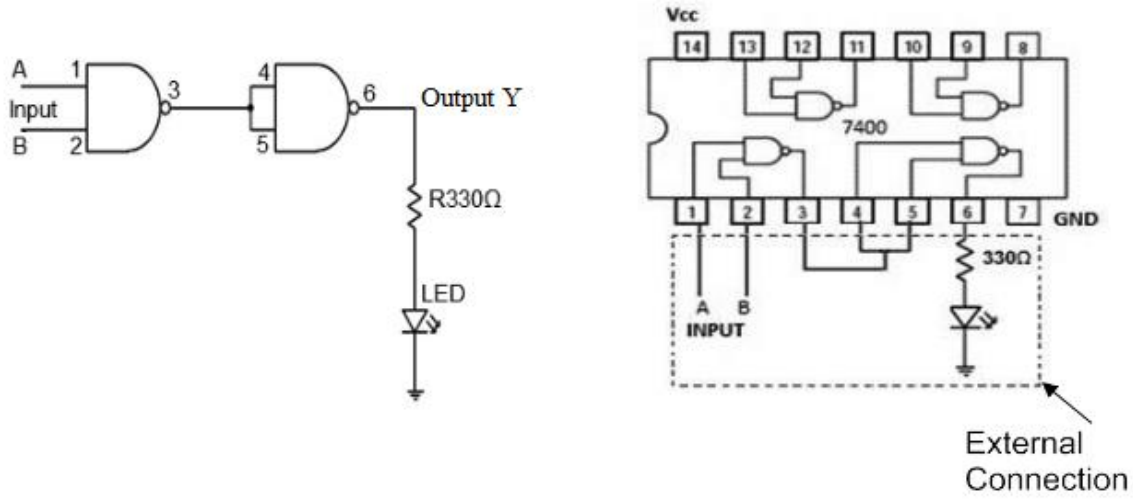


Figure 2.5 Logic diagram and IC Circuit diagram to verify AND operation using NAND gate IC 7400.

OR Gate:

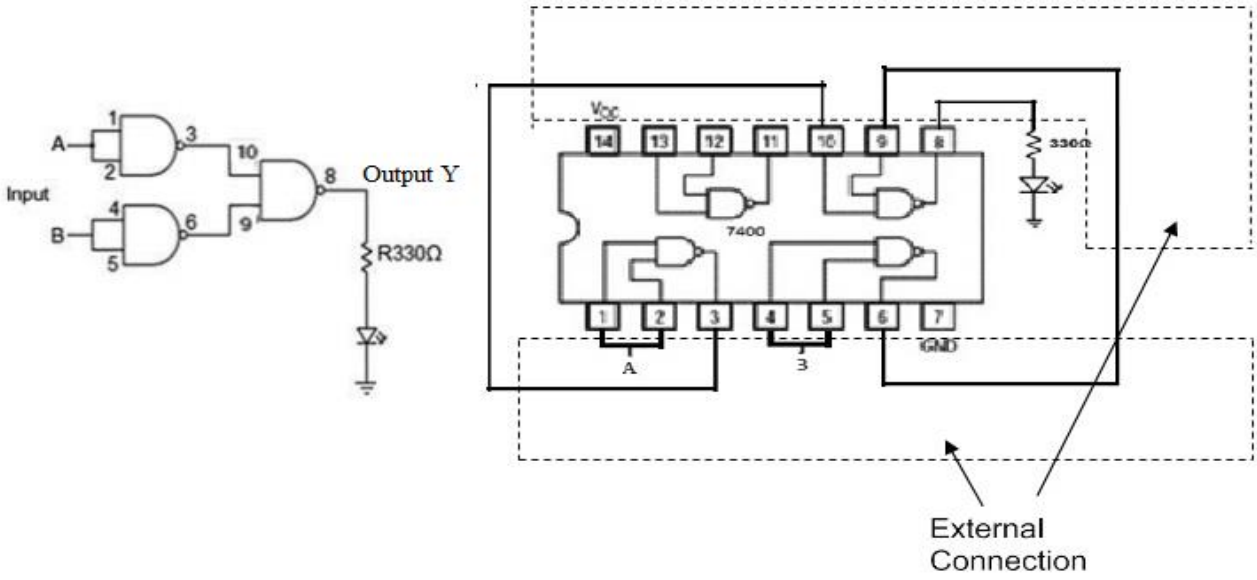


Figure 2.6 Logic diagram and IC Circuit diagram to verify OR operation using NAND gate IC 7400.

NOT Gate:

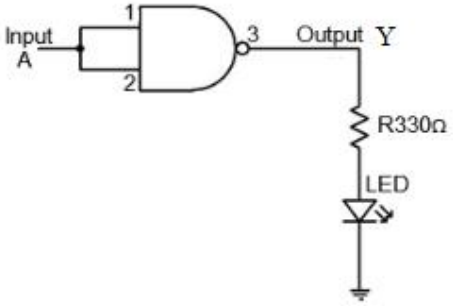


Figure (a)

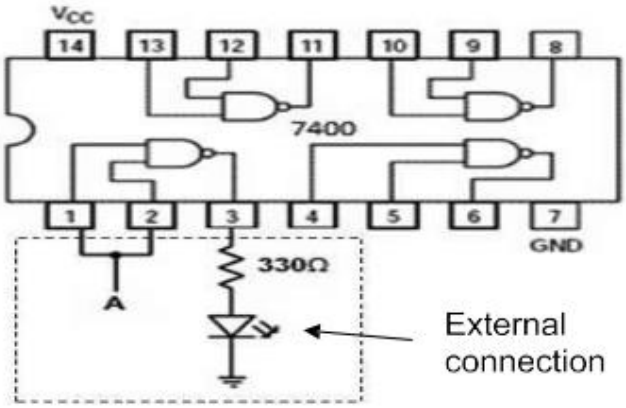


Figure (b)

Figure 2.7 Logic diagram and IC Circuit diagram to verify NOT operation using NAND gate IC 7400.

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5cm X 17 cm.	1
5	IC	7400 or any IC of equivalent function..	1
6	LED	Red/Yellow color 5 mm.	1
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	1

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of all ICs used.
5. Make the connections as per the logic diagram given in Figure 2.5.
6. Apply logic inputs and verify the truth table for each gate by observing LED status for output.
7. Measure the output voltage with the help of a digital multimeter.
8. Write the reading in each observation table.
9. Repeat the process for all logic functions by making the connections as shown in Figure 2.6 and 2.7.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table of AND, OR gates using NAND (Universal) Gate IC.

Sr. No.	Inputs		AND Gate		OR Gate	
	A	B	LED Status (ON/Off)	Output Voltage (V)	LED Status (ON/Off)	Output Voltage (V)
1.	0	0				
2.	0	1				
3.	1	0				
4.	1	1				

Table 2: Verification of the truth table of NOT gates using NAND (Universal) Gate IC.

Sr. No.	Inputs	NOT Gate	
	A	LED Status (ON/Off)	Output Voltage (V)
1.	0		
2.	1		

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Design digital circuit for the following equation using NAND gates only.

$$Y=(\overline{\overline{A}\cdot\overline{B}})+(\overline{C\cdot D})+\overline{E}$$

2. Design digital circuit for EX-OR and EX-NOR Gates using NAND gates only.
3. Identify the IC number which is compatible for IC 7400 (NAND Gate IC), Write the part numbering and specifications of that IC.

[Space for Answers]

……

……

……

……

XVIII References/Suggestions for further reading

1. https://de-iitr.vlabs.ac.in/exp/realization-of-logic-functions/
2. https://www.electrical4u.com/universal-gate-nand-nor-gate-as-universal-gate/
3. https://drive.google.com/file/d/1qqF_ycwfPO1v_4wyXKLTc-N_xYOPUlak/view?amp;invite= CMS0y-wJ&ts=5f0ef639
4. https://learnabout-electronics.org/Digital/dig21.php#:~:text=Seven%20Basic%20Logic%20Gates&text=Digital%20electronics%20relies%20on%20the,(Exclusive%20NOR)%20and%20NOT.

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.3: Building of AND, OR, NOT gates using NOR gate.

I Practical Significance

NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate. So its output is a complement of the output of an OR gate. This gate can have a minimum of two inputs; output is always one. By using only NOR gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NAND. So this gate is also called as universal gate

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Apply knowledge of number system and logic circuits in working of digital system.

IV Laboratory Learning Outcome

Build AND, OR, NOT gates using NOR gate and verify NOR gate as universal gate.

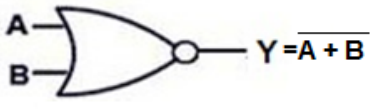
V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

NAND and NOR gates are universal gates because any Boolean expression can be realized using these gates. The NOR gate has the ability to perform three operations such as AND, OR and NOT. This gate is a combination of OR and NOT gates. The NOR gate output is equal to the inverse of the OR gate. The symbol and truth table is as shown in figure 3.1.

Logic Symbol



Truth Table

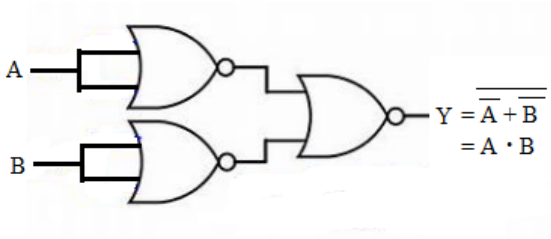
Input A	Input B	Output Y
0	0	1
0	1	0
1	0	0
1	1	0

Figure 3.1 Logic symbol and truth table of NOR gate

NOR gate as AND gate:

From DeMorgan's theorems, a bubbled NOR gate is equivalent to the AND gate.

AND Gate using NOR Gates



Truth Table of AND Gate

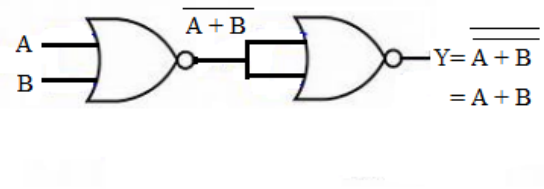
Input A	Input B	Output Y
0	0	0
0	1	0
1	0	0
1	1	1

Figure 3.2 Connection diagram for NOR gate as AND gate.

NOR gate as OR gate:

A NOR produces a complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.

OR Gate using NOR Gates



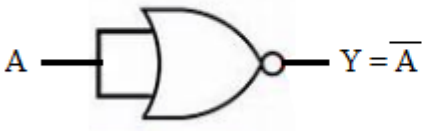
Truth Table of OR Gate

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 3.3 Connection diagram for NOR gate as OR gate.

NOR gate as NOT gate: A NOT produces complement of the input. It can have only one input, tie the inputs of a NOR gate together. Now it will work as a NOT gate.

NOT Gate using NOR Gate



Truth Table of NOT Gate

Input A	Output Y
0	1
1	0

Figure 3.4 Connection diagram for NOR gate as NOT gate.

VII Actual Circuit diagram used in laboratory with related equipment rating-

A) Suggestive Circuit diagram-

AND Gate:

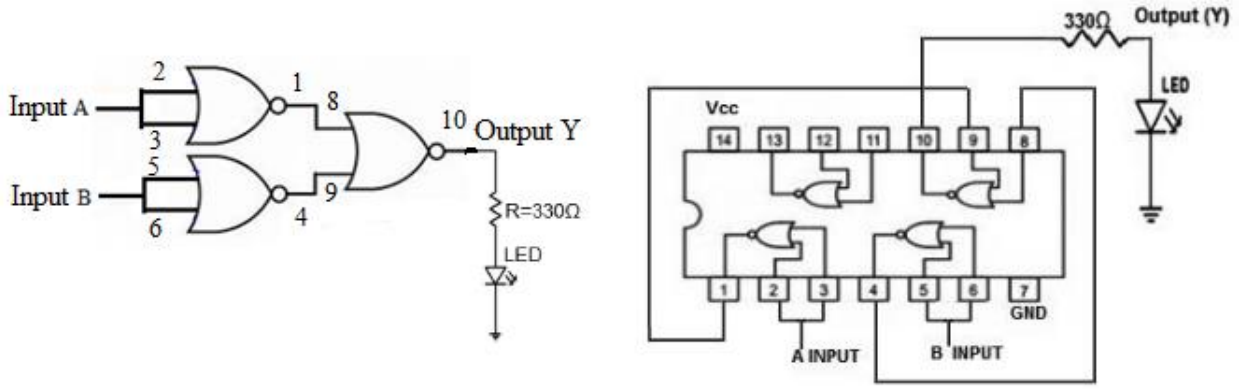


Figure 3.5 Logic diagram and IC Circuit diagram to verify AND operation using NOR gate IC 7402.

OR Gate:

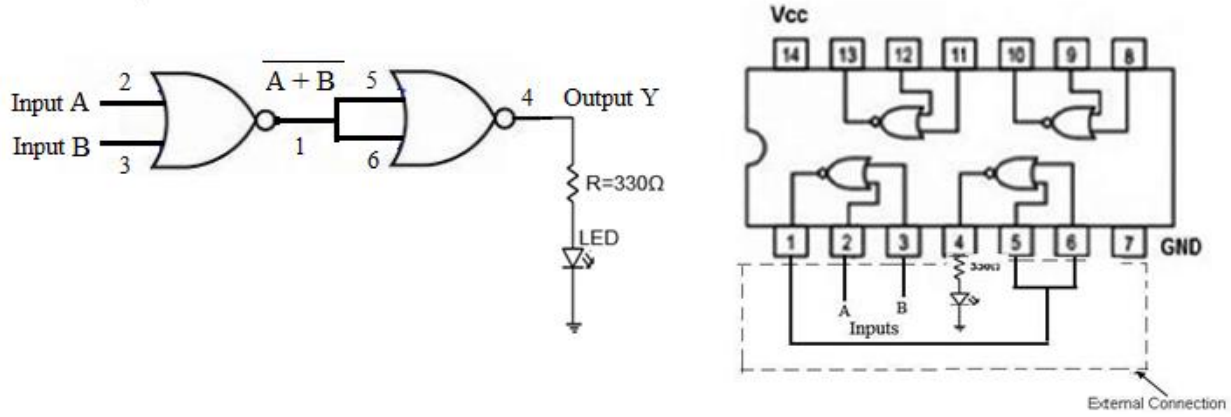


Figure 3.6 Logic diagram and IC Circuit diagram to verify OR operation using NOR gate IC 7402.

NOT Gate:

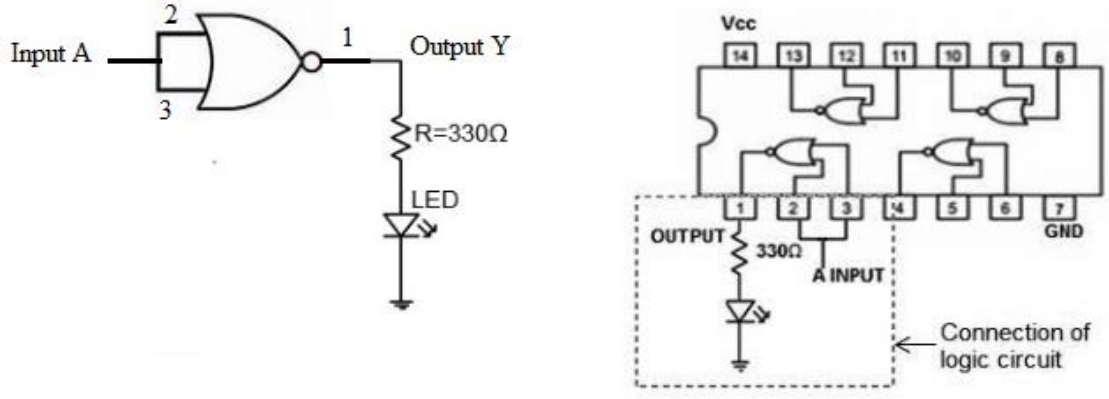


Figure 3.7 Logic diagram and IC Circuit diagram to verify NOT operation using NOR gate IC 7402.

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5cm X 17 cm.	1
5	IC	7402 or any IC of equivalent function.	1
6	LED	Red/Yellow color 5 mm.	1
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	1

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of all ICs used.
5. Make the connections as per the logic diagram given in Figure 3.5.
6. Apply logic inputs and verify the truth table for each gate by observing LED status for output.
7. Measure the output voltage with the help of a digital multimeter.
8. Write the reading in each observation table.
9. Repeat the process for all logic functions by making the connections as shown in Figure 3.6 and 3.7.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table of AND, OR gates using NOR (Universal) Gate IC.

Sr. No.	Inputs		AND Gate		OR Gate	
	A	B	LED Status (ON/Off)	Output Voltage (V)	LED Status (ON/Off)	Output Voltage (V)
1.	0	0				
2.	0	1				
3.	1	0				
4.	1	1				

Table 2: Verification of the truth table of NOT gates using NOR (Universal) Gate IC.

Sr. No.	Input	NOT Gate	
	A	LED Status (ON/Off)	Output Voltage (V)
1.	0		
2.	1		

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Design digital circuit for NAND Gate using NOR gates only.
2. Write name of manufacturers of digital IC7402.
3. Identify the IC number, which is compatible for IC 7402 (NOR Gate IC), Write the part numbering and significance of the letters used for part numbering.
4. Identify and state the front panel controls of an IC tester used in your Laboratory.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://de-iitr.vlabs.ac.in/exp/realization-of-logic-functions/
2. https://www.electrical4u.com/universal-gate-nand-nor-gate-as-universal-gate/
3. https://drive.google.com/file/d/1qqF_ycwfPO1v_4wyXKLTc-N_xYOPUlak/view?amp;invite= CMS0y-wJ&ts=5f0ef639

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.4: Building of Half adder and Half subtractor using Boolean expressions.

I Practical Significance

Adder and subtractor are the combinational logic circuits used to perform basic arithmetic operations like addition and subtraction of two binary digits. They are used in ALUs of computers and other kinds of processors. They are also used in other parts of the processor to calculate addresses, table indices, and increment and decrement operations. Half adder and subtractor can be used only for two bit addition and subtraction respectively.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Apply knowledge of number system and logic circuits in working of digital system.

IV Laboratory Learning Outcome

Build Half adder and Half subtractor.

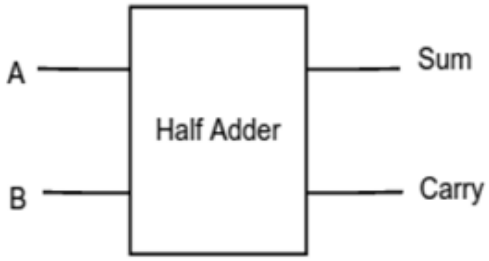
V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

A half-adder is composed of one X-OR gate and one AND gate that produces two binary outputs from two binary inputs. It adds two one-bit binary numbers “A” and “B”. The output is the sum of the two bits “SUM” and the carry “CARRY”. The C output is 1 only when both inputs are 1. The S output represents the least significant bit of the sum.

Logic Diagram



Truth Table

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 4.1 Logic symbol and truth table of Half adder

Logic equations for outputs of Half adder are,

$\text{Sum} = A \oplus B$

$\text{Carry} = A \cdot B$

So, Half adder circuit contains an Ex-OR gate to get Sum output. To get Carry output, input A and input B are given to the AND gate.

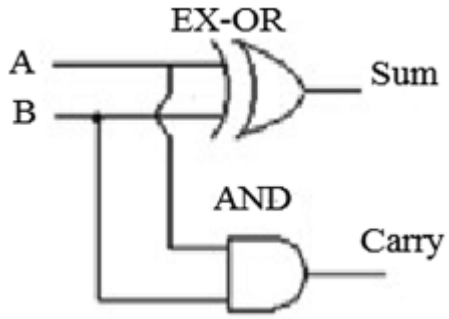
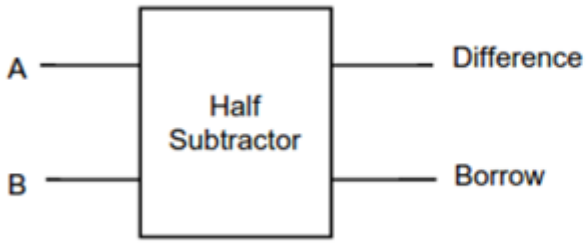


Figure 4.2 Connection diagram for Half adder

A half subtractor is a combinational circuit that subtracts two bits. It has two inputs A and B and two outputs Difference and Borrow. A half-subtractor is a combinational circuit, which is used to perform subtraction of two bits. It has two inputs, A (minuend) and B (subtrahend) and two outputs D (difference) and B(borrow). It is constructed using X-OR gate, NOT gate (Inverter), and AND gate. The B output is 1 only when the subtrahend (B) is greater than the minuend (A).

Logic Diagram



Truth Table

Inputs		Output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Figure 4.3 Logic symbol and truth table of Half subtractor

Logic equations for outputs of Half subtractor are,

$$\text{Sum} = A \oplus B$$

$$\text{Carry} = \overline{A} \cdot B$$

So, Half subtractor circuit contains an Ex-OR gate to get Difference output. To get Borrow output, input A is inverted and then input B and inverted A input is given to the AND gate.

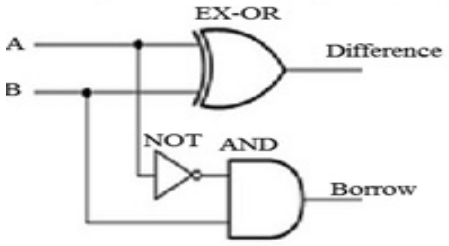


Figure 4.4 Connection diagram for Half subtractor

VII Actual Circuit diagram used in laboratory with related equipment rating-

A) Suggestive Circuit diagram-

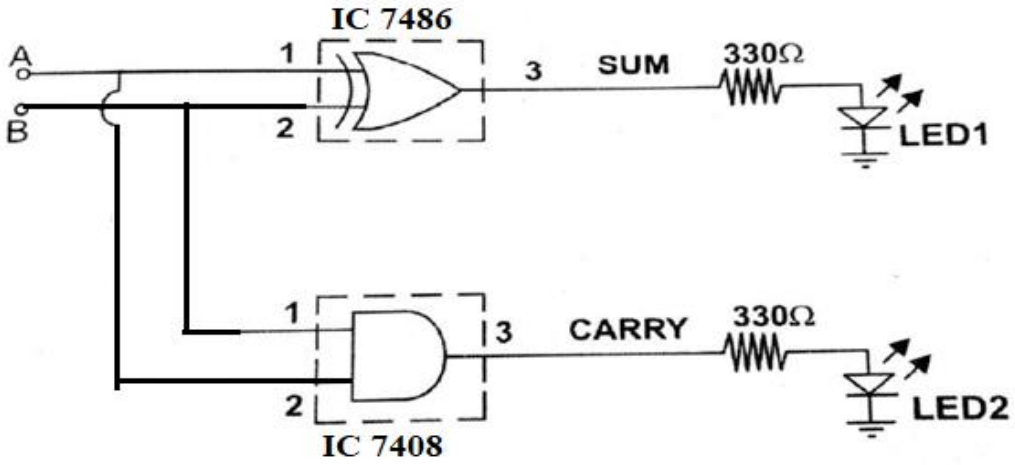


Figure 4.5 Half-adder circuit

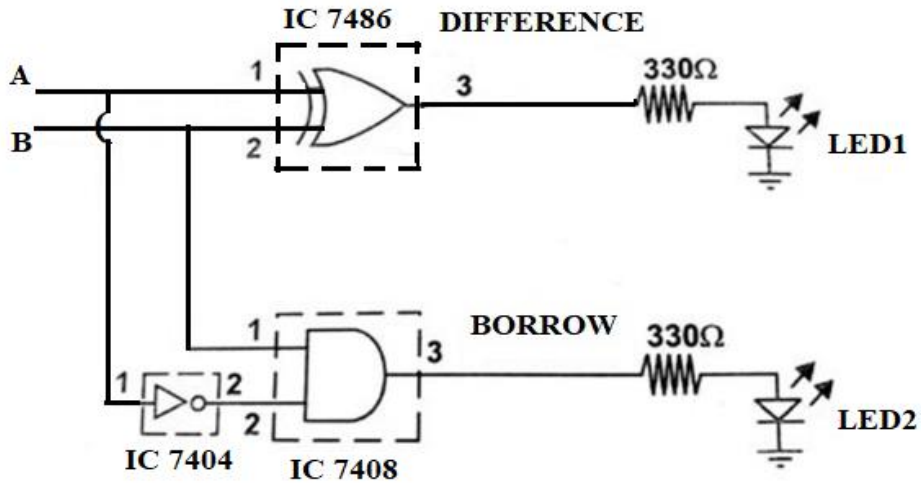


Figure 4.6 Half-subtractor circuit

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5cm X 17 cm.	1
5	IC	7486, 7408 and 7404 or any ICs of equivalent function.	1
6	LED	Red/Yellow color 5 mm.	1
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of all ICs used.
5. Make the connections as per the circuit diagram given in Figure 4.5 for Half adder and figure 4.6 for Half subtractor respectively.
6. Apply logic inputs and verify the truth table for half adder as well as half subtractor by observing LED status for output.
7. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table for Half adder.

Sr. No.	Inputs		Output of Half adder	
	A	B	Sum	Carry
1.	0	0		
2.	0	1		
3.	1	0		
4.	1	1		

Table 2: Verification of the truth table for Half subtractor.

Sr. No.	Inputs		Output of Half subtractor	
	A	B	Difference	Borrow
1.	0	0		
2.	0	1		
3.	1	0		
4.	1	1		

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Give any two applications of half adder and half subtractor.
2. Draw half adder using NAND gates only.
3. Draw half subtractor using NOR gates only.
4. State the features of NAND and NOR gate ICs used to construct Half adder circuit.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://de-iitr.vlabs.ac.in/digital-electronics-iitr/exp/half-full-adder/
2. https://www.watelectronics.com/what-is-half-adder-circuit-diagram-its-applications/#google_vignette
3. https://www.gatevidyalay.com/half-adder/
4. https://www.gatevidyalay.com/half-subtractor/
5.https://www.watelectronics.com/what-is-half-adder-circuit-diagram-its-applications/#google_vignette

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.5: Building of Full adder and full subtractor using Boolean expressions.

I Practical Significance

Digital computers perform a variety of information tasks. Among the functions encountered are the various arithmetic operations. The most basic arithmetic operation is the addition or subtraction of binary digits. A binary adder-subtractor is a combinational circuit that performs the arithmetic operations of addition and subtraction with binary numbers. They are used in ALUs of many computers and other kinds of processors. In this practical, students will build a circuit of full adder and full subtractor to perform addition and subtraction of 3 bits.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Apply knowledge of number system and logic circuits in working of digital system.

IV Laboratory Learning Outcome

Build Full adder and full subtractor.

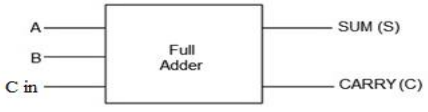
V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

Full adder-The difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. Full adder is used to perform the addition of multiple bit numbers. The first two inputs are A and B and the third input is an input carry as Cin, generated from previous stage bit addition and outputs are sum(S) and carry(C). Full adder is used to perform addition of three input bits.

Logic Symbol



Truth Table of Full adder

INPUTS			OUTPUTS	
A	B	C_{in}	SUM	CARRY OUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 5.1 Logic symbol and Truth table of Full adder

Logic equations for outputs of Full adder are,

$$\text{Sum} = A \oplus B \oplus C$$

$$\text{Carry} = A \cdot B + A \cdot C + B \cdot C$$

So, Full adder circuit contains two 2-input Ex-OR gates (IC 7486) to get Sum output and three 2-input AND gates (IC 7408) and two 2-input OR gates (IC 7432) to get Carry output.

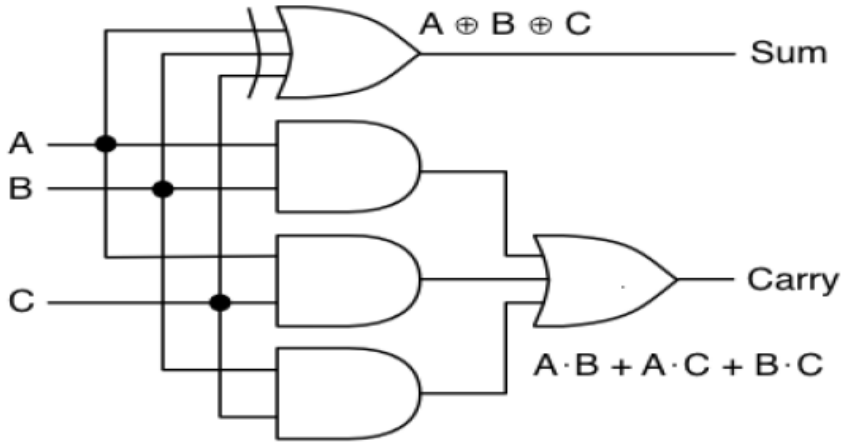
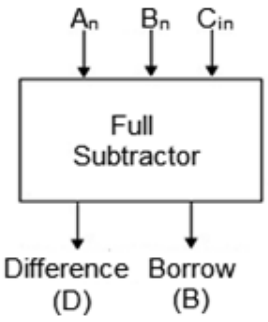


Figure 5.2 Connection diagram for Full adder

Full subtractor- Full subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and other is subtrahend. In full subtractor '1' is borrowed by the previous adjacent lower minuend bit. Hence these three (A, B, Cin) bits are considered as the input of a full subtractor. There are two outputs, that are DIFFERENCE output D and BORROW output B. It has to take care of repeated borrow by the next higher bit.

Logic Symbol



Truth Table of Full subtractor

Inputs			Outputs	
A	B	C	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Figure 5.3 Logic symbol and truth table of Full subtractor

Logic equations for outputs of Full subtractor are,

$$\text{Difference} = A \oplus B \oplus C$$

$$\text{Carry} = \overline{A} \cdot B + \overline{A} \cdot C + B \cdot C$$

So, Full subtractor circuit contains two 2-input Ex-OR gates (IC 7486) to get Difference output and one NOT gate (IC 7404), three 2-input AND gates (IC 7408) and two 2-input OR gates (IC 7432) to get Borrow output.

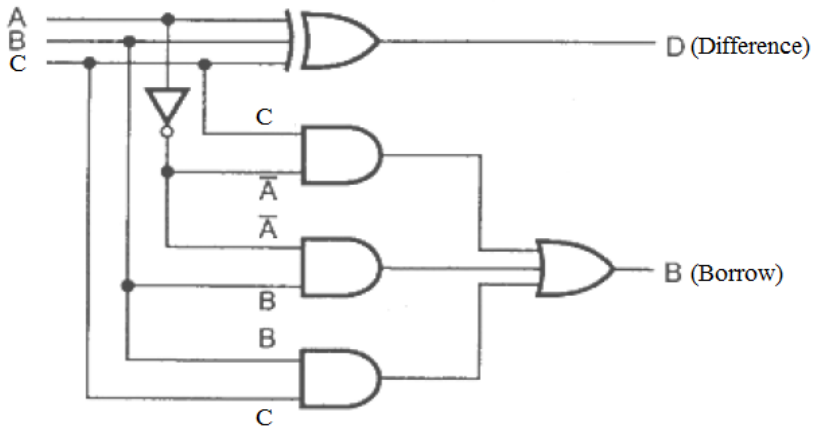


Figure 5.4 Connection diagram for Full subtractor

Pin out of the ICs required to connect the circuit:

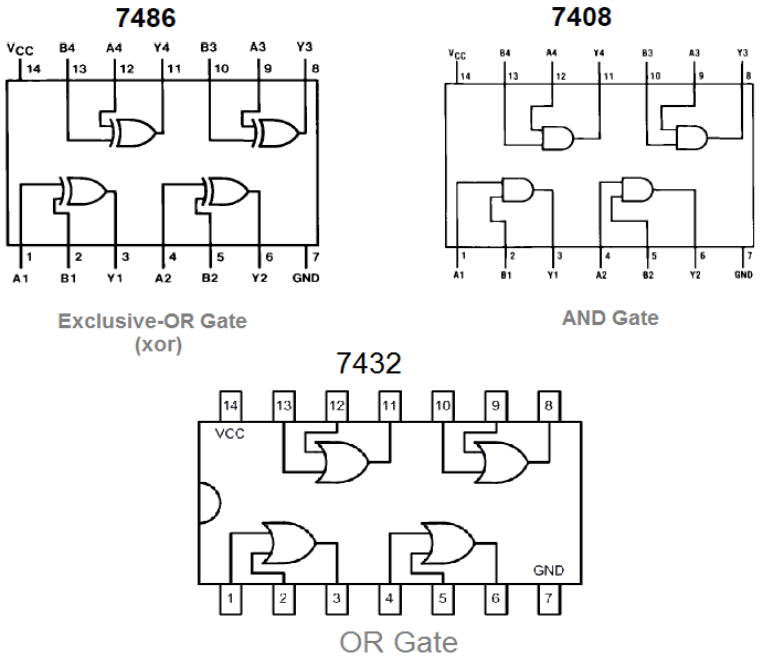
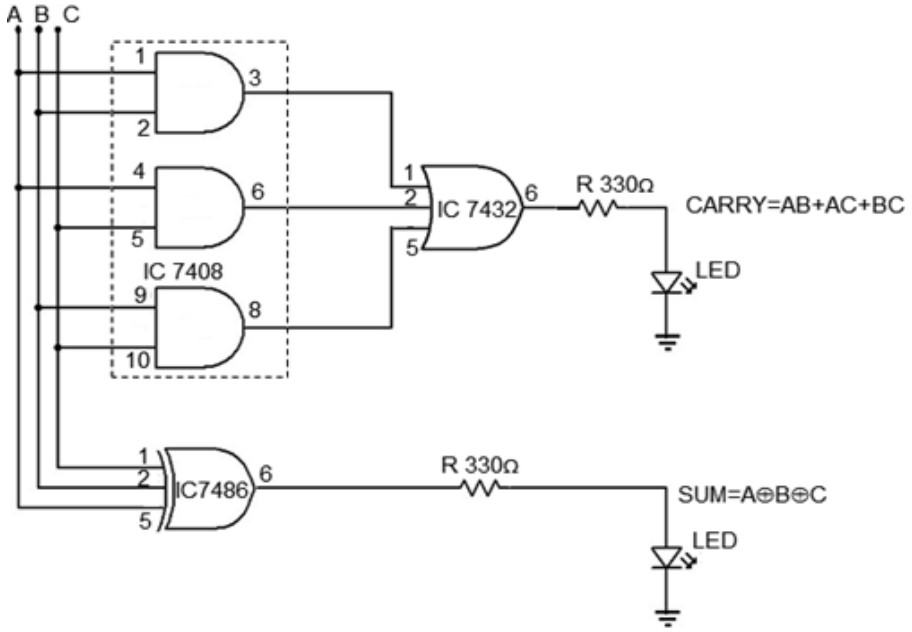


Figure 5.5 Pin out of the ICs required for connecting the circuit

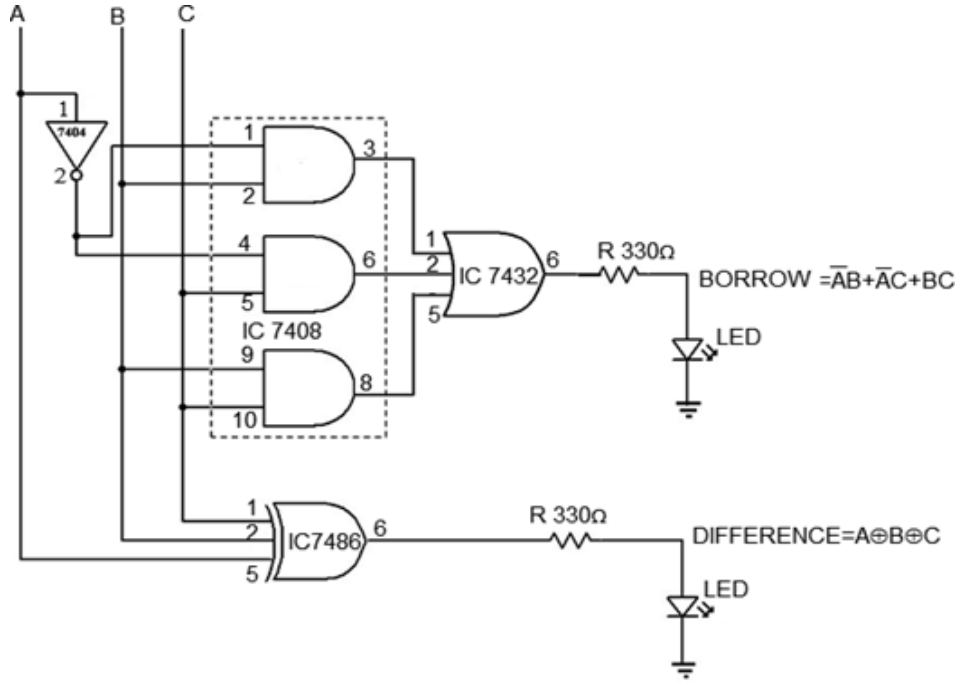
VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-



Note: Pin No. 3 and Pin No. 4 of IC 7432 are shorted to make 3 input OR gate and Pin No. 3 and Pin No. 4 of IC 7486 are shorted to make 3 input Ex-OR gate.

Figure 5.6 Full adder circuit



Note: Pin No. 3 and Pin No. 4 of IC 7432 are shorted to make 3 input OR gate and Pin No. 3 and Pin No. 4 of IC 7486 are shorted to make 3 input Ex-OR gate.

Figure 5.7 Full subtractor circuit

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5cm X 17 cm.	1
5	IC	7486, 7432, 7408 and 7404 or any ICs of equivalent function.	1
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition, then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of all ICs used.
5. Make the connections as per the circuit diagram given in Figure 5.5 for Full adder and figure 5.6 for Full subtractor respectively.
6. Apply logic inputs and verify the truth table for Full adder as well as Full subtractor by observing LED status for output.
7. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table for Full adder.

Sr. No.	Inputs			Output of Full adder	
	A	**B**	**C**	**Sum**	**Carry**
1.	0	0	0		
2.	0	0	1		
3.	0	1	0		
4.	0	1	1		
5.	1	0	0		
6.	1	0	1		
7.	1	1	0		
8.	1	1	1		

Table 2: Verification of the truth table for Full subtractor.

Sr. No.	Inputs			Output of Full subtractor	
	A	**B**	**C**	**Difference**	**Borrow**
1.	0	0	0		
2.	0	0	1		
3.	0	1	0		
4.	0	1	1		
5.	1	0	0		
6.	1	0	1		
7.	1	1	0		
8.	1	1	1		

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Give any two applications of full adder and full subtractor.
2. Give IC numbers for 3-input Ex-OR gate and 3-input OR gate.
3. Draw a digital circuit diagram of full adder using two half adders.
4. Draw a digital circuit diagram of full adder using only NAND gates.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://de-iitr.vlabs.ac.in/digital-electronics-iitr/exp/half-full-adder/
2. https://drive.google.com/file/d/1E70Fqx8lGJKrh2TDx8cYWmSLHztjS2ni/view
3. https://www.gatevidyalay.com/full-adder/
4. https://www.gatevidyalay.com/full-subtractor/

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.6: Verification of operation of Multiplexer IC74151

I Practical Significance

In most of the electronic systems, the digital data is available on more than one line. It is necessary to route this data over a single line. Under such circumstances we require a circuit which selects one of the many inputs at a time. This circuit is a multiplexer (MUX), which has many inputs, one output and some select inputs. After verification of the truth table of the multiplexer, students can improve hands-on skills to use multiplexers and can understand where and how the multiplexer is used and how the multiplexer improves the reliability of the digital system as it reduces the number of external wired connections.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Build a Multiplexer using IC.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

Multiplexer (MUX)is a combinational circuit that is one of the most widely used in digital design. The multiplexer is a data selector which gets one out of several inputs to a single output. It has **n** data inputs and one output line and **m** select lines where $\mathbf{2^m = n}$ shown in figure 6.1. Depending upon the digital code applied at the select inputs one out of n data input is selected and transmitted to a single output channel. Normally strobe or enable (E) input is incorporated which is generally active low which enables the multiplexer when it is LOW. IC 74151A is an 8: 1 multiplexer which provides two complementary outputs Y and $\overline{Y}$. The output Y is the same as the selected input and $\overline{Y}$ is its complement. The n : 1 multiplexer can be used to realize m variable function. ($\mathbf{2^m = n}$, m is no. of select inputs). For example, to design 8:1 multiplexer, eight data inputs are required (i.e. n=8) and three select lines are required (i.e. m=3).

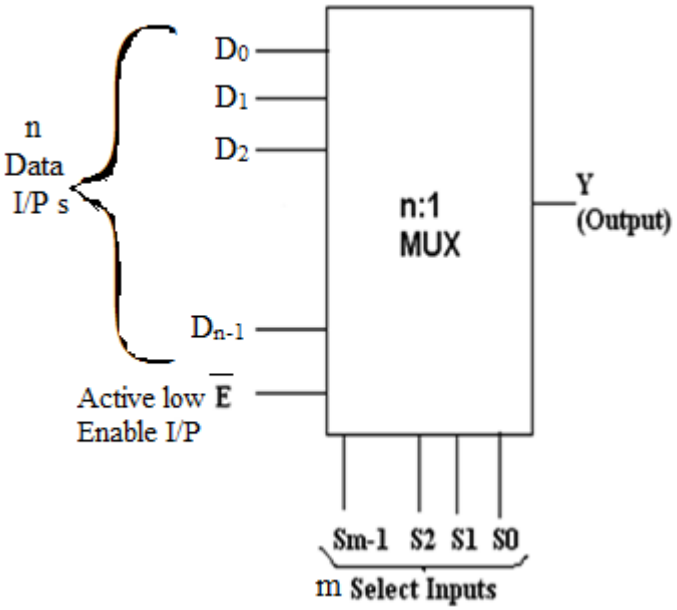


Figure 6.1 Logic diagram of n:1 Multiplexer

According to the number of data inputs, different types of multiplexers are given below,

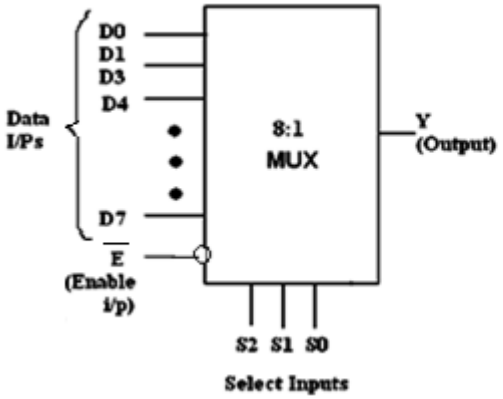
1. 2:1 MUX (2 inputs and 1 output)
2. 4:1 MUX (4 inputs and 1 output)
3. 8:1 MUX (8 inputs and 1 output)
4. 16:1 MUX (2 inputs and 1 output)

Some of the available Multiplexer ICs and their functions and output state are given below,

IC No.	Function of the IC	Output state of the IC
IC 74157	Quad 2:1 MUX	Output same as input given
IC 74158	Quad 2:1 MUX	Output is inverted input
IC 74153	Dual 4:1 MUX	Output same as input given
IC 74152	Dual 4:1 MUX	Output is inverted input
IC 74151A	8:1 MUX	Both outputs are available (Y & $\overline{Y}$)
IC 74151	8:1 MUX	Output is inverted input
IC 74150	16:1 MUX	Output is inverted input

Table 6.1 Available Multiplexer ICs and their functions and output state

Logic diagram & Truth table of 8:1 Multiplexer



Enable	Select Inputs			Output
$\overline{E}$	S2	S1	S0	Y
1	X	X	X	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	0	0	0	D4
0	0	0	1	D5
0	0	1	0	D6
0	0	1	1	D7

Figure 6.2 Logic diagram and truth table of 8:1 multiplexer

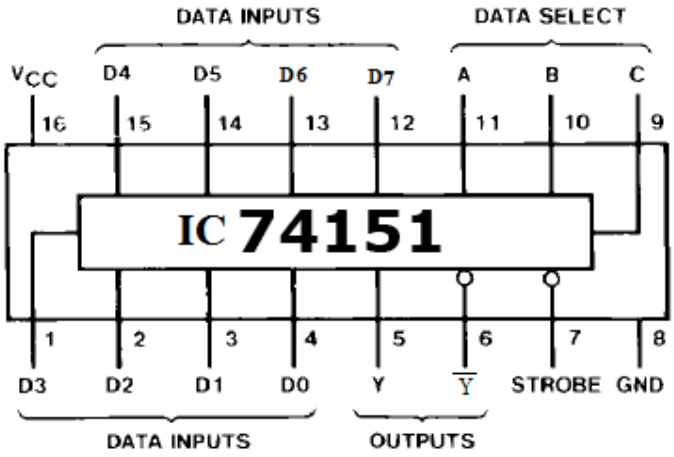


Figure 6.3 Pin out of the 8:1 Multiplexer IC-74151

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

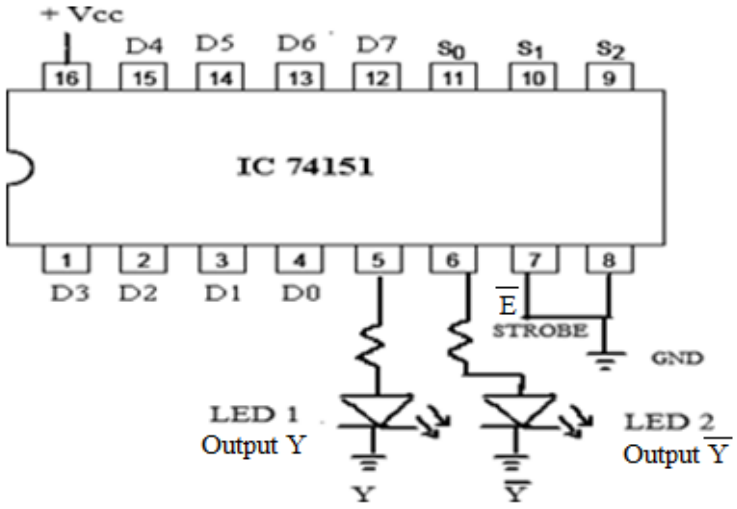


Figure 6.3 Circuit diagram for 8:1 Multiplexer

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	74151	1
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Make the connections as per the circuit diagram given in Figure 6.3.
6. Apply logic inputs at data inputs and select inputs and verify the truth table for 8:1 multiplexer by observing LED status for output at pin no.5 and complementary output at pin no. 6.
7. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table for Multiplexer.

Inputs					Outputs	
Strobe (Enable) I/P	Data input	Select inputs			Y	$\overline{Y}$
$\overline{E}$	D_n	S2	S1	S0		
0		0	0	0		
0		0	0	1		
0		0	1	0		
0		0	1	1		
0		1	0	0		
0		1	0	1		
0		1	1	0		
0		1	1	1		
1		X	X	X		

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Design 8:1 multiplexer using basic gates.
2. Write the specifications of IC 74151 from the data sheet.
3. Draw MUX tree for 16:1 MUX using 8:1 MUXs.
4. List the name of manufacturers of the multiplexer ICs used in your laboratory.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. **https://www.alldatasheet.com/datasheet-pdf/pdf/7831/NSC/74151.html**
2. https://drive.google.com/file/d/1z9zBUp6Rnt6i3Ie5_z4x2T0_ARqBCyam/view
3. https://de-iitr.vlabs.ac.in/exp/multiplexer-demultiplexer/theory.html
4. https://de-iitr.vlabs.ac.in/exp/multiplexer-demultiplexer/theory.html

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.7: Verification of operation of Demultiplexer IC74155.

I Practical Significance

A demultiplexer (or demux) is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. An electronic demultiplexer can be considered as a single-input, multiple-output switch. Demultiplexers are mainly used in Boolean function generators and decoder circuits. After verification of the truth table of demultiplexer, students can improve hands-on skills and can understand where and how to use demultiplexers.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Build a Demultiplexer using IC.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

Demultiplexer has only one input and "n" number of outputs along with "m" number of select inputs. A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line. Hence demultiplexer is equivalent to a single pole multiple way switch as shown in figure 7.1. The enable input will enable the demultiplexer. The relation between the n output lines and m select lines is as given below for 1 : n DEMUX,

$$\mathbf{n = 2^m}$$

The demultiplexer performs the opposite process to a multiplexing process; it performs "one to many" operation. It has only one input (D) and n number of outputs (Y0, Y1, Y2... Yn-1) as shown in the figure 7.1 given below. Demultiplexer can also be used as a decoder e.g. Binary to Decimal Decoder. Data input given is Logic level 1 or 0, active low or active high strobe/enable pin is used for enabling DEMUX

For example, to design a 1:8 multiplexer, one data input and eight output lines are required (i.e. n=8) and three select lines are required (i.e. m=3).

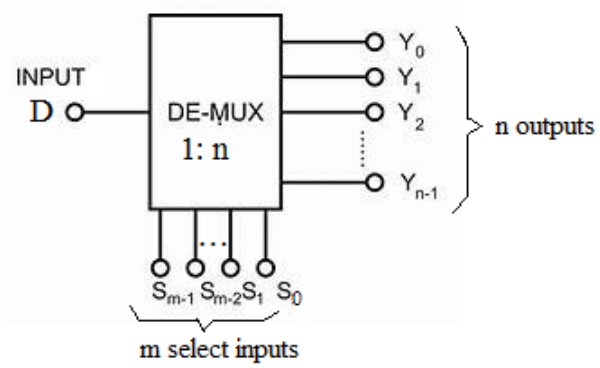


Figure 7.1 Logic diagram of 1:n Demultiplexer

According to the number of data inputs, different types of multiplexers are given below,

1. 1:2 DEMUX (1 input and 2 outputs)
2. 1:4 DEMUX (1 input and 4 outputs)
3. 1:8 DEMUX (1 input and 8 outputs)
4. 1:16 DEMUX (1 input and 16 outputs)

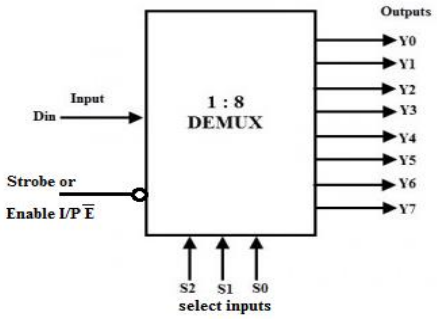


Figure 7.2 Logic diagram of 1:8 Demultiplexer

Some of the available Demultiplexer ICs and their functions and output state are given below,

Available Demultiplexer ICs and their functions and output state

IC No.	Function of the IC	Output state of the IC
IC 74139	Dual 1:4 demux	Output is inverted input
IC 74156	Dual 1:4 demux	Output is open collector
IC 74138	1:8 demux	Output is inverted input
IC 74238	1:8 demux	Output is same as input
IC 74155	Dual 1:4 demux or 1:8 demux	1. For 1st 1:4 demux output is inverted input 2. For 2nd 1:4 demux output is same as input 3. For 1:8 demux strobe is used as data input, which is active low and output is same as input
IC 74154	1:16 demux	Output is same as input
IC 74159	1:16 demux	Output is open collector and same as input

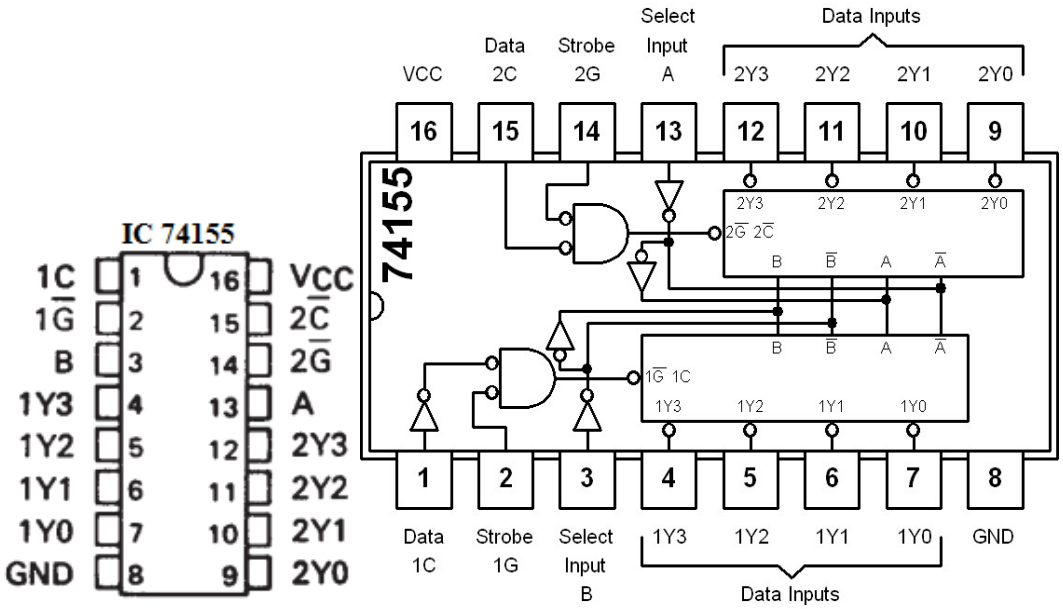
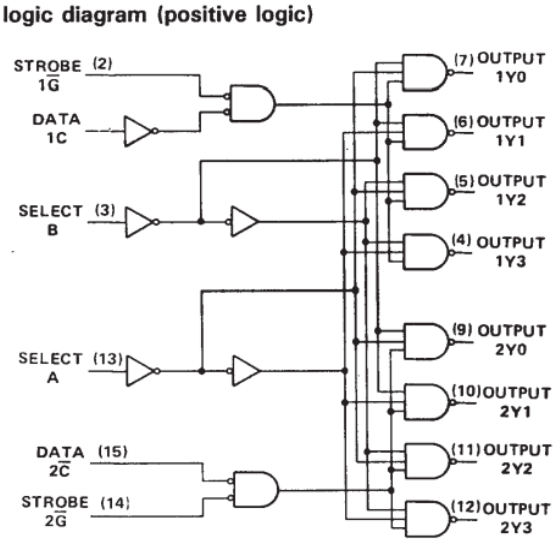


Figure 7.3 Pim diagram of DEMUX IC 74155

The function of IC 74155 is dual 1:4 demultiplexer. It can be used as 1:8 demultiplexer IC by connecting Data lines of both the demultiplexers to each other and used as a third select input. Strobe inputs of both the 1:4 demultiplexers are connected together and used as data input.



Truth Table
2-LINE-TO-4-LINE DECODER
OR 1-LINE-TO-4-LINE DEMULTIPLEXER

INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	1G	1C	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	2G	2C	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

Figure 7.4 Logic diagram and truth table of 1:4 demultiplexers of IC 74155

Truth Table

3-LINE-TO-8-LINE DECODER

OR 1-LINE-TO-8-LINE DEMULTIPLEXER

INPUTS				OUTPUTS							
SELECT			STROBE OR DATA	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
C†	B	A	$\overline{G}$‡	2Y0	2Y1	2Y2	2Y3	1Y0	1Y1	1Y2	1Y3
X	X	X	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H
H	H	L	L	H	H	H	H	H	H	L	H
H	H	H	L	H	H	H	H	H	H	H	L

†C = inputs 1C and 2$\overline{C}$ connected together

‡$\overline{G}$ = inputs 1$\overline{G}$ and 2$\overline{G}$ connected together

H = high level, L = low level, X = irrelevant

Figure 7.5 Truth table of 1:8 demultiplexer of IC 74155

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

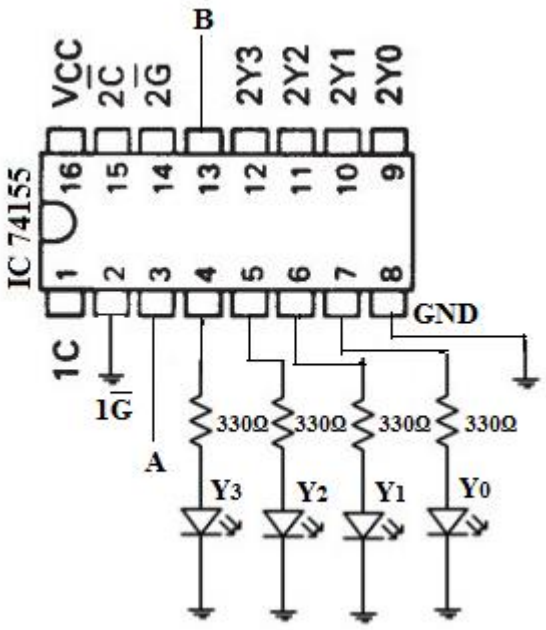


Figure 7.6 Circuit diagram for first 1:4 Demultiplexer of IC74155

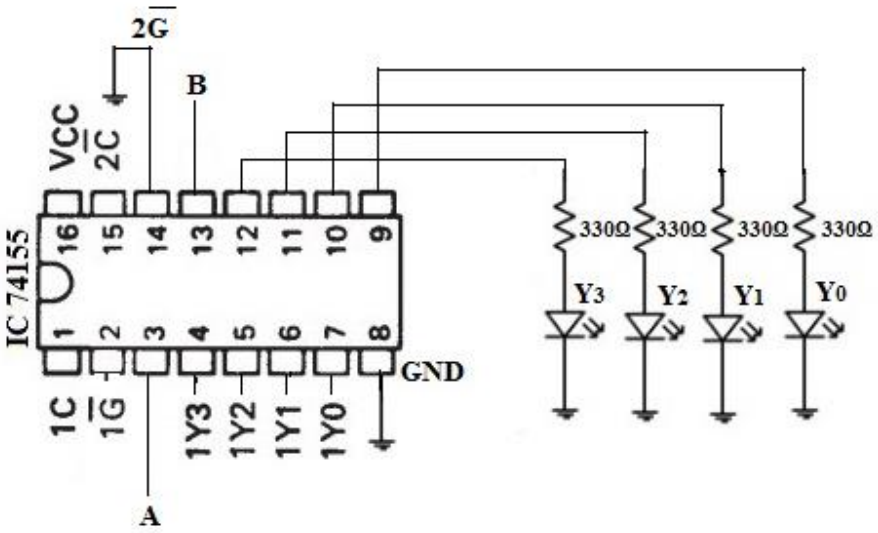


Figure 7.7 Circuit diagram for second 1:4 Demultiplexer of IC74155

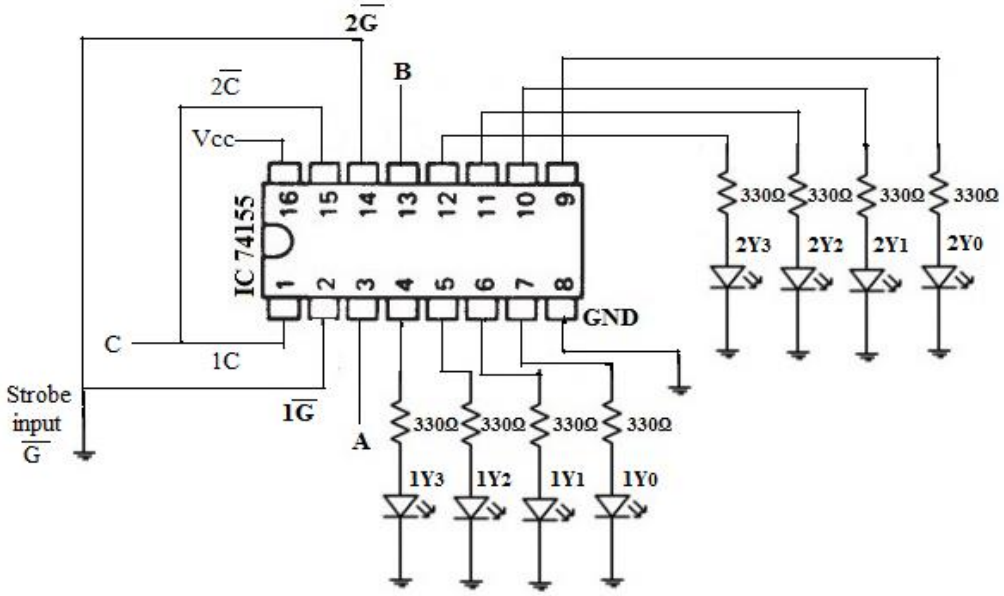


Figure 7.8 Circuit diagram for 1:8 Demultiplexer of IC74155

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	74155	1
6	LED	Red/Yellow color 5 mm.	8
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	8

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition, then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Make the connections as per the circuit diagram given in Figure 7.6.
6. Apply logic inputs at data inputs and select inputs. Verify the truth table for the first 1:4 demultiplexer of IC74155 by observing LED status for outputs.
7. Write the reading in each observation table.
8. Repeat the procedure for the second 1:4 demultiplexer and 1:8 multiplexer of IC74155 by making the connections as shown in figure 7.7 and 7.8.
9. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table for first 1:4 Demultiplexer.

Inputs				Outputs			
Select Inputs		Strobe	Data				
B	A	$1\overline{G}$	1C	$1Y_0$	$1Y_1$	$1Y_2$	$1Y_3$
X	X	1	X				
0	0	0	1				
0	1	0	1				
1	0	0	1				
1	1	0	1				
X	X	X	0				

Table 2: Verification of the truth table for second 1:4 Demultiplexer.

Inputs				Outputs			
Select Inputs		Strobe	Data				
B	A	$2\overline{G}$	$2\overline{C}$	$2Y_0$	$2Y_1$	$2Y_2$	$2Y_3$
X	X	1	X				
0	0	0	0				
0	1	0	0				
1	0	0	0				
1	1	0	0				
X	X	X	1				

Table 3: Verification of the truth table for 1:8 Demultiplexer.

Inputs				Outputs							
Select Inputs			Strobe or Data	0	1	2	3	4	5	6	7
C	B	A	$\overline{G}$	$2Y_0$	$2Y_1$	$2Y_2$	$2Y_3$	$1Y_0$	$1Y_1$	$1Y_2$	$1Y_3$
X	X	X	1								
0	0	0	0								
0	0	1	0								
0	1	0	0								
0	1	1	0								
1	0	0	0								
1	0	1	0								
1	1	0	0								
1	1	1	0								

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Draw the logic diagram of 1:8 DEMUX using basic gates.
2. Draw the logic diagram of 1:8 DEMUX using NAND gates only.
3. Write the specifications of IC 74151 from the data sheet.
4. Write the equivalent CMOS MUX IC numbers.
5. Draw a DEMUX tree for 1:16 DEMUX using 1:8 DEMUXs.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://drive.google.com/file/d/1NVqK_3Yq-LFlDf_9bS0q80uF3n4W8Uuf/view
2. https://de-iitr.vlabs.ac.in/exp/multiplexer-demultiplexer/theory.html
3. https://www.ti.com/lit/ds/symlink/sn74ls155a.pdf?ts=1717609035375&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN74LS155A
4. https://en.wikipedia.org/wiki/Multiplexer

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.8: Testing the function of RS flip flop using NAND Gate.

I Practical Significance

The flip flop is a one bit memory cell that stores one bit of information. The data available in memory can be used for further operation. The flip flops are used as registers in microcontrollers or processors. It is also used in random access memory, binary counters, delay elements and bounce elimination switches. As NAND gate is an universal gate RS flip-flop can be designed using NAND gates only. By doing this practical students can improve their psychomotor skills and can understand the concept of changing the output state by changing the input signals and clock signal condition applied to the sequential logic circuits.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Test the function of RS flipflop.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

Digital circuits have many combinations of logic circuits. They are classified as either combinational or sequential. The output of combinational circuits depends only on the present inputs. In contrast, in sequential logic circuit output depends on the present state of input as well as past output.

A flip-flop is a bi-stable circuit i.e. it has two stable states. It stores one bit of information and it is the basic building block of semiconductor memory. The FFs change their output state depending upon inputs at a certain interval of time synchronized with some clock pulse applied to it.

Types of flip flops

1. SR-FF (Set Reset flip flop) or RS-FF
2. JK-FF (JK flip flop)
3. MSJK-FF (Master Slave JK flip flop)
4. D-FF (Delay flip flop)
5. T-FF (Toggle flip flop)

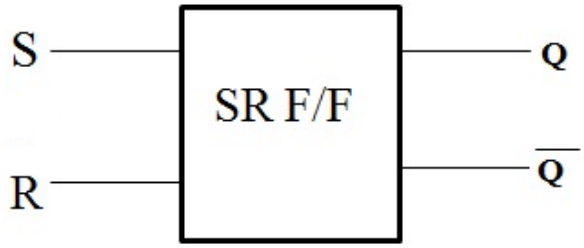


Figure 8.1 Logic diagram of SR Flip-flop

Inputs		Outputs		Remark
S	**R**	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	Q_n	$\overline{Q}_n$	No Change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Forbidden State

*where Q_n is present state output and $\overline{Q}_{n+1}$ is next state output.

Figure 8.2 Truth Table of SR flip-flop

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

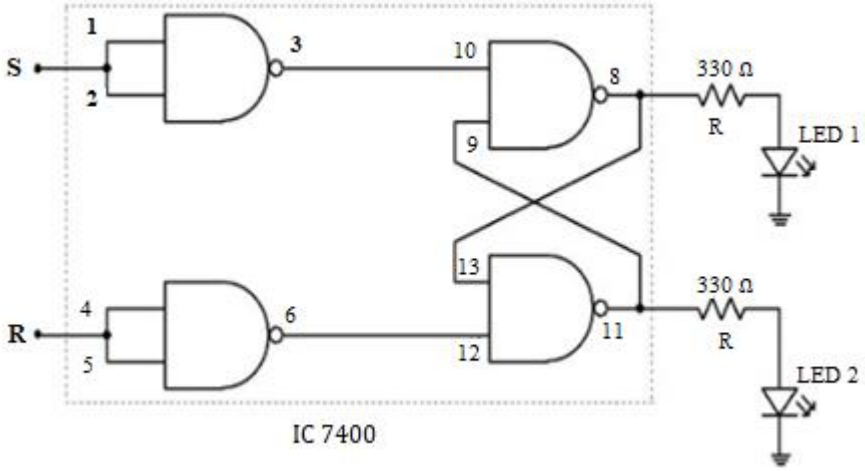


Figure 8.3 Circuit diagram of SR flip-flop

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	7400	1
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Make the connections as per the circuit diagram given in Figure 8.3.
6. Apply logic inputs to S and R inputs and verify the truth table of SR flip-flop given in figure 8.2.
7. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table of SR Flip-flop

Inputs		Outputs		Remark
S	**R**	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0			
0	1			
1	0			
1	1			

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. List the available ICs having the function of SR flip-flop.
2. Draw the digital circuit for SR flip-flop using NOR only.
3. Design D-flip-flop using SR flip-flop. Verify the truth table of D-FF.

[Space for Answers]

……

……

……

……

……

……

……

……

XVIII References-

1. http://vlabs.iitkgp.ac.in/dec/exp8/exp8a/exp8p1.html
2. https://de-iitr.vlabs.ac.in/exp/truth-tables-flip-flops/procedure.html
3. https://drive.google.com/file/d/1GNCefxZBGZA56B3wstcOfBdXNLswKi_Y/view
4. https://www.electronics-tutorials.ws/sequential/seq_1.html

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.9: Testing the function of JK flip flop using 7476.

I Practical Significance

The limitation of SR flip flop is overcome in JK flip flop. Forbidden state which is undefined state in SR flip flop is eliminated in JK flip flop. In JK flip flop when J=K=1, the output is in toggle state, which is uncertain if clock input changes its state and J and K inputs remain at logic high for a longer period than clock pulse period, this situation is called Race around condition. To avoid the problem of race around condition the JK flip flop in Master and slave mode is used.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Test the function of JK flip flop.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

Master Slave J K flip flop is a cascade of two J-K flip-flops, with feedback from the outputs of the second flip flop to the inputs of the first. The first part is called a master flip-flop while the next is called a slave flip-flop. Here the external clock pulse triggers the master flip-flop while the slave is activated at its inversion i.e. if the master is positive-level triggered, then the slave is negative-level triggered and vice-versa. This means that the data enters into the flip-flop at positive or negative level of the clock pulse while it is obtained at the output pins during positive or negative level of the clock pulse. Hence, a master-slave flip-flop completes its operation only after the appearance of one full clock pulse. For example, on the positive transition of the clock, the data from the J and K inputs is transferred to the master. On the negative transition of the clock, the data from the master is transferred to the slave.

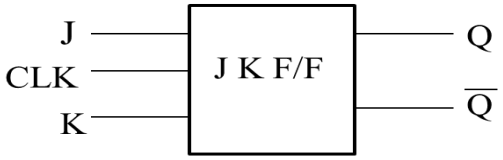


Figure 9.1 Logic diagram of JK Flip-flop

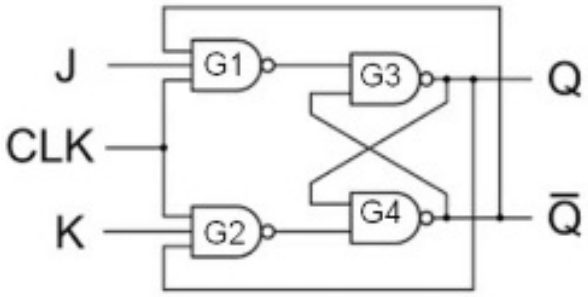


Figure 9.2 Circuit diagram of JK Flip-flop using NAND gates

Inputs		**Outputs**		**Remark**
J	**K**	**Q_{n+1}**	**$\overline{Q}_{n+1}$**	
0	0	Qn	$\overline{Q}_n$	No Change
0	1	0	1	Reset
1	0	1	0	Set
1	1	$\overline{Q}_n$	Qn	Toggle state

*where Qn is present state output and $\overline{Q}_{n+1}$ is next state output.

Figure 9.3 Truth Table of JK flip-flop

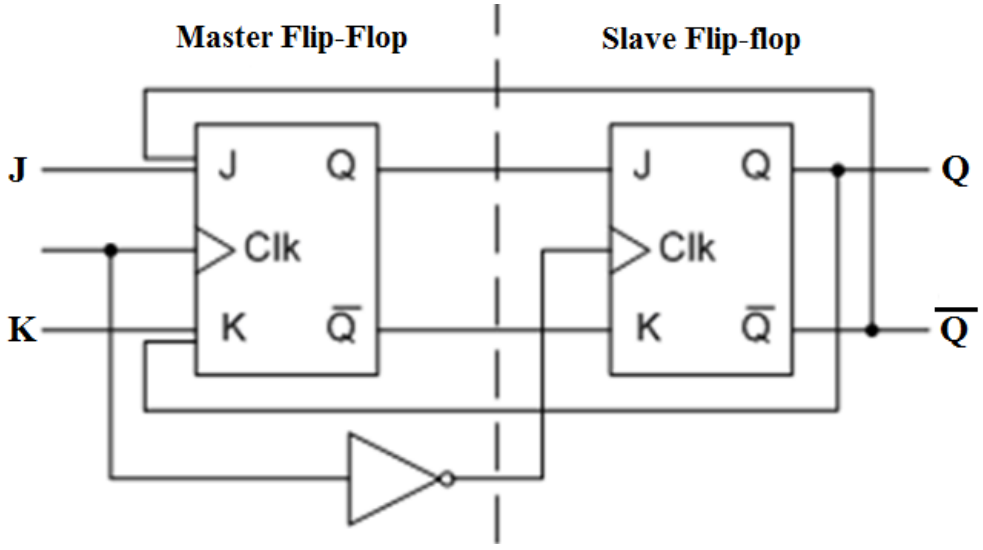


Figure 9.4 Logic diagram of Master-slave JK flip-flop

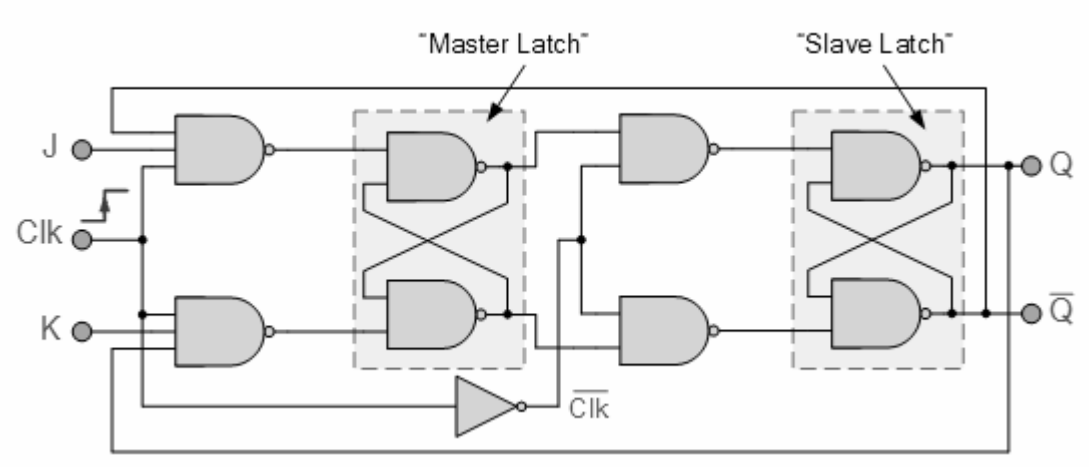


Figure 9.5 Master-slave JK flip-flop using NAND gates

IC7476- IC7476 is a Dual Master-slave JK flip-flop IC. It has two negative edge triggered JK flip-flops with active low preset and clear inputs. Preset (PR) input should be low to set the output Q (logic level 1). Clear (CR or CLR) input should be low to clear the output Q (logic level 0).

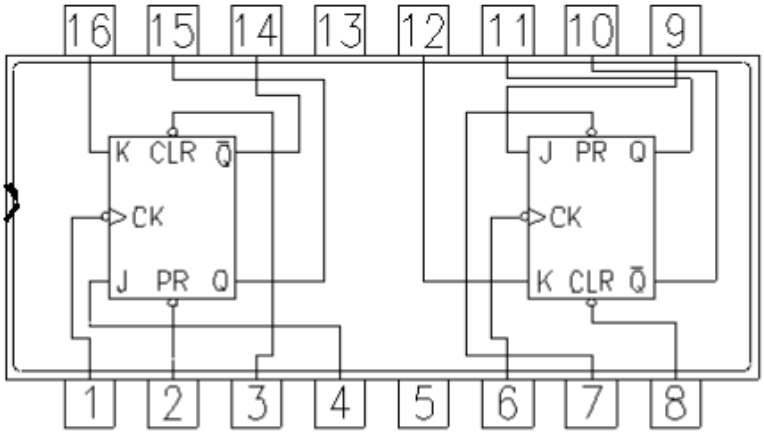


Figure 9.6 IC-7476 Dual Master-slave JK flip-flop with Preset and Clear inputs

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

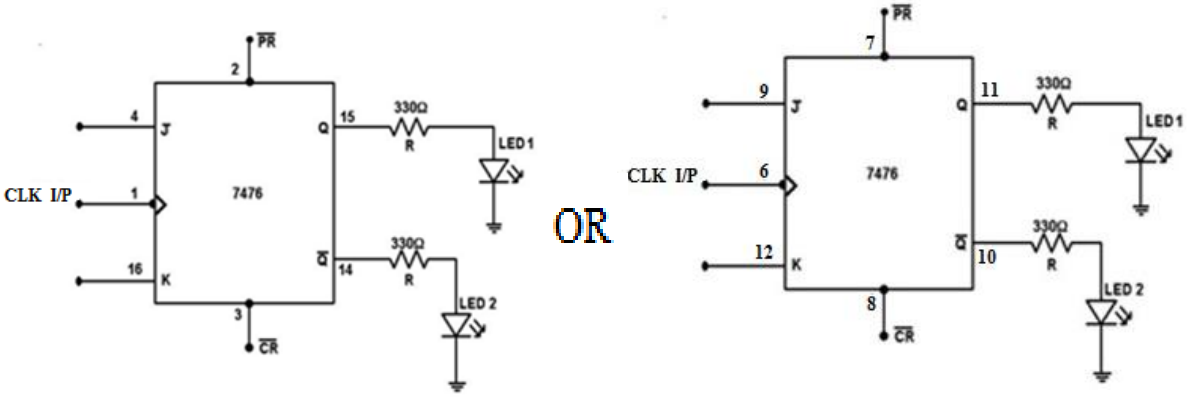


Figure 9.7 Circuit diagram of Master slave JK flip-flop using IC 7476

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	7476	1
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Study the pin diagram of IC 7476 carefully.
6. Make the connections as per the circuit diagram given in Figure 9.7.
7. Apply logic inputs to J and K inputs and verify the truth table of JK flip-flop as per the theoretical output.
8. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table of Master Slave JK Flip-flop using IC 7476

Sr. No.	$\overline{PR}$	$\overline{CR}$	Jn	Kn	Clk	Theoretical O/P		Practical O/P	
						Qn	$\overline{Q}_n$	Qn	$\overline{Q}_n$
1.	0	1	X	X	X	1	0		
2.	1	0	X	X	X	0	1		
3.	1	1	0	0	X	Previous Output			
4.	1	1	X	X	0	Previous Output			
5.	1	1	1	0	↓	1	0		
6.	1	1	0	1	↓	0	1		
7.	1	1	1	1	↓	Toggle			

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. How many MS JK flip flops are there in IC7476? Refer to the data sheet and write the specifications of IC 7476.
2. How JK flip-flop can be used as frequency divider
3. Design D and T flip-flops using JK flip-flop. Verify the truth table of D-FF and T-FF.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References-

1. https://www.alldatasheet.com/datasheet-pdf/pdf/50913/FAIRCHILD/7476.html
2. https://www.futurlec.com/74/IC7476.shtml
3. https://www.electronics-tutorials.ws/sequential/seq_2.html
4. https://nptel.ac.in/courses/108105132
5. https://drive.google.com/file/d/1YYEvVpB0-6xcmaJT1_eO9m8jacFBjTfW/view
6. https://de-iitr.vlabs.ac.in/exp/truth-tables-flip-flops/theory.html

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.10: Construction and testing the functionality of D flip flop using IC 7476.

I Practical Significance

IC 7476 is a 16 pin dual negative edge triggered JK flip-flop IC. It can be used as D or T type flip-flops. D flip-flops are the basic building blocks of registers. D flip flop can be designed using JK or SR flip flop.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Construct and test the functionality of D flip flop.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

IC7476 is a Dual Master-slave JK flip-flop IC. It has two negative edge triggered JK flip-flops with active low preset and clear inputs. Preset (PR) input should be low to set the output Q (logic level 1). Clear (CR or CLR) input should be low to clear the output Q (logic level 0).The truth table of JK flip-flop is as given in figure 10.3.

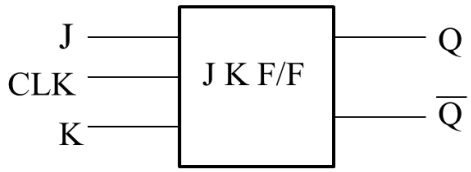


Figure 10.1 Logic diagram of JK Flip-flop

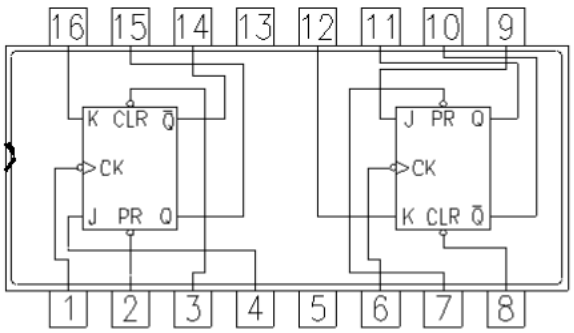


Figure 10.2 IC-7476 Dual Master-slave JK flip-flop with Preset and Clear inputs

Sr. No.	$\overline{PR}$	$\overline{CR}$	Jn	Kn	Clk	Q_n	$\overline{Q}_n$
1.	0	1	X	X	X	1	0
2.	1	0	X	X	X	0	1
3.	1	1	0	0	X	Previous Output	
4.	1	1	X	X	0	Previous Output	
5.	1	1	1	0	↓	1	0
6.	1	1	0	1	↓	0	1
7.	1	1	1	1	↓	Toggle	

Table 10.1 Truth Table of JK flip-flop

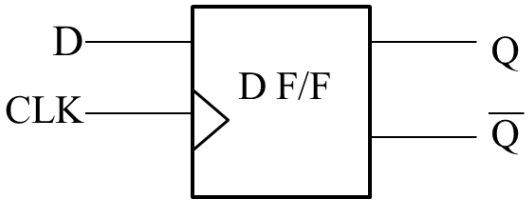


Figure 10.3 Logic diagram of D flip-flop

The D flip-flop is obtained from the JK/SR flip-flop by connecting a NOT gate between the J/S and K/R inputs of the JK/SR flip-flop and T flip-flop is obtained from JK flip-flop by shorting J and K inputs of the JK flip-flop.

Table 10.2: Truth Table of D flip-flop

Input	**Output**		**Remark**
D	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	1	Reset
1	1	0	Set

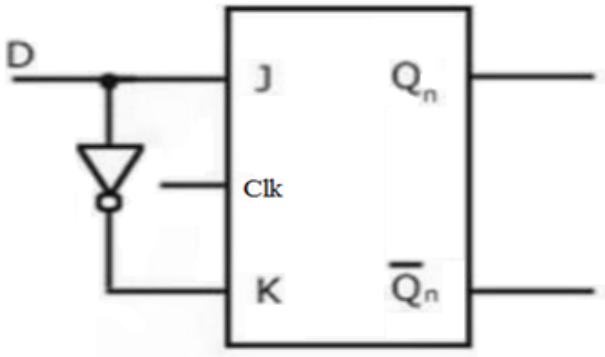


Figure 10.4 JK Flip Flop converted into D Flip flop

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

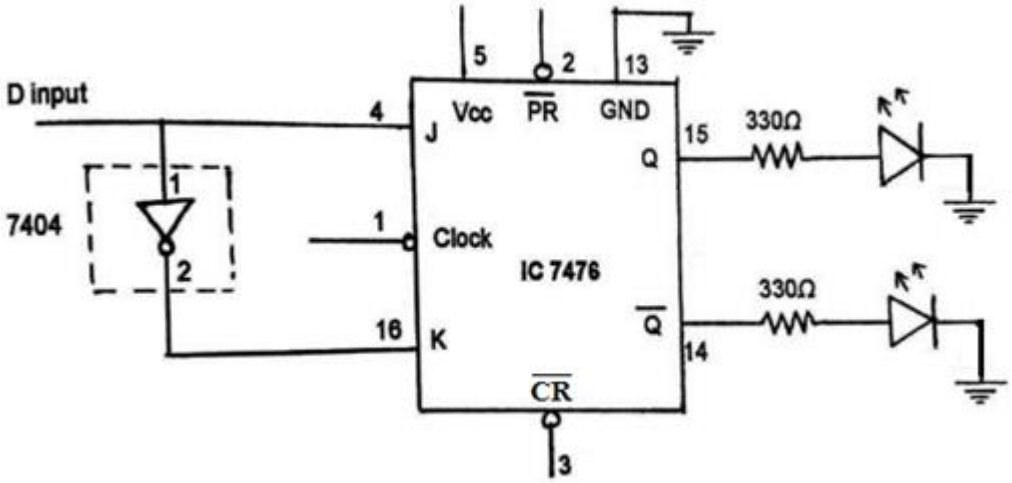


Figure 10.5 Circuit diagram of D flip-flop using IC 7476

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	7476, 7404	1 each
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition, then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Study the pin diagram of IC 7476 carefully.
6. Make the connections to convert JK flip flop to D flip flop as per the circuit diagram given in Figure 10.5.
7. Apply logic input to D input and verify the truth table of D flip-flop as per the theoretical output.
8. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table of D Flip-flop using IC 7476

Sr. No.	$\overline{PR}$	$\overline{CR}$	D	Clk	Theoretical O/P		Practical O/P	
					Q_n	$\overline{Q_n}$	Q_n	$\overline{Q_n}$
1.	0	1	X	X	1	0		
2.	1	0	X	X	0	1		
3.	1	1	0	X	Previous output			
4.	1	1	X	0	Previous output			
5.	1	1	1	↓	1	0		
6.	1	1	0	↓	0	1		

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Compare D flip flop using IC 7476 and IC 7474 by referring to data sheets.
2. Give two applications of D flip flop.
3. Draw and explain how the D flip flop can be used as a frequency divider. (Hint: use two D flip flops)
4. Refer to the data sheet of D-FF ICs from TTL and CMOS logic families and compare the propagation delay for both the logic families.

[Space for Answers]

……

……

……

……

……

XVIII References-

1. https://www.alldatasheet.com/datasheet-pdf/pdf/50913/FAIRCHILD/7476.html
2. https://www.futurlec.com/74/IC7476.shtml
3. https://www.electronics-tutorials.ws/sequential/seq_2.html
4. https://nptel.ac.in/courses/108105132
5. https://drive.google.com/file/d/1YYEvVpB0-6xcmaJT1_eO9m8jacFBjTfW/view

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.11: Construction and testing the functionality of T flip flop using IC 7476.

I Practical Significance

IC 7476 is a 16 pin dual negative edge triggered JK flip-flop IC. It can be used as D or T type flip-flops. D Flip-Flop (Delay Flip-Flop) is used to provide time delay. They are basic building blocks of Shift Registers T Flip-Flop (Toggle Flip-Flop) is used in counters. It can also be used as a frequency divider. T flip flop can be designed using JK flip flop but it can not be designed by using SR flip flop.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Construct and test the functionality of T flip flop.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

IC7476 is a Dual Master-slave JK flip-flop IC. It has two negative edge triggered JK flip-flops with active low preset and clear inputs. Preset (PR) input should be low to set the output Q (logic level 1). Clear (CR or CLR) input should be low to clear the output Q (logic level 0). The truth table of JK flip-flop is as given in figure 11.3.

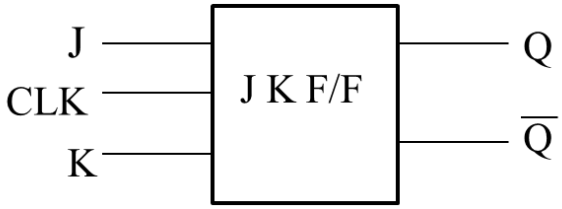


Figure 11.1 Logic diagram of JK Flip-flop

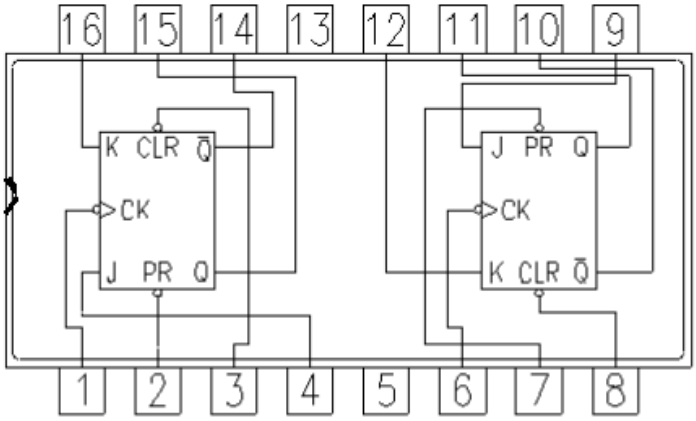


Figure 11.2 IC-7476 Dual Master-slave JK flip-flop with Preset and Clear inputs

Sr. No.	$\overline{PR}$	$\overline{CR}$	Jn	Kn	Clk	Qn	$\overline{Q}_n$
1.	0	1	X	X	X	1	0
2.	1	0	X	X	X	0	1
3.	1	1	0	0	X	Previous Output	
4.	1	1	X	X	0	Previous Output	
5.	1	1	1	0	↓	1	0
6.	1	1	0	1	↓	0	1
7.	1	1	1	1	↓	Toggle	

Figure 11.3 Truth Table of JK flip-flop

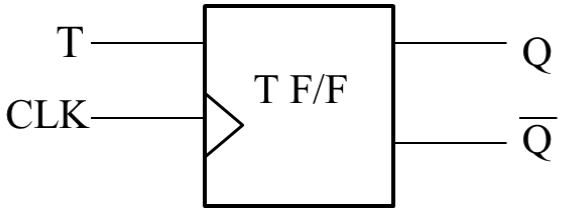


Figure 11.4 Logic diagram of T flip-flop

The D flip-flop is obtained from the JK/SR flip-flop by connecting a NOT gate between the J/S and K/R inputs of the JK/SR flip-flop and T flip-flop is obtained from JK flip-flop by shorting J and K inputs of the JK flip-flop.

Input	**Output**		**Remark**
T	$\mathbf{Q_{n+1}}$	$\overline{Q}_{n+1}$	
0	Qn	$\overline{Q}_n$	Previous state output
1	$\overline{Q}_n$	Qn	Toggle state output

Figure 11.5 Truth Table of T flip-flop

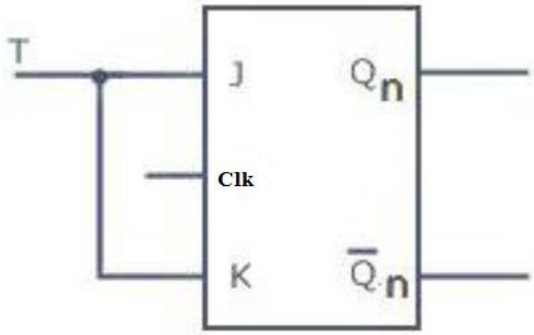


Figure 10.6 JK Flip Flop converted into T Flip flop

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

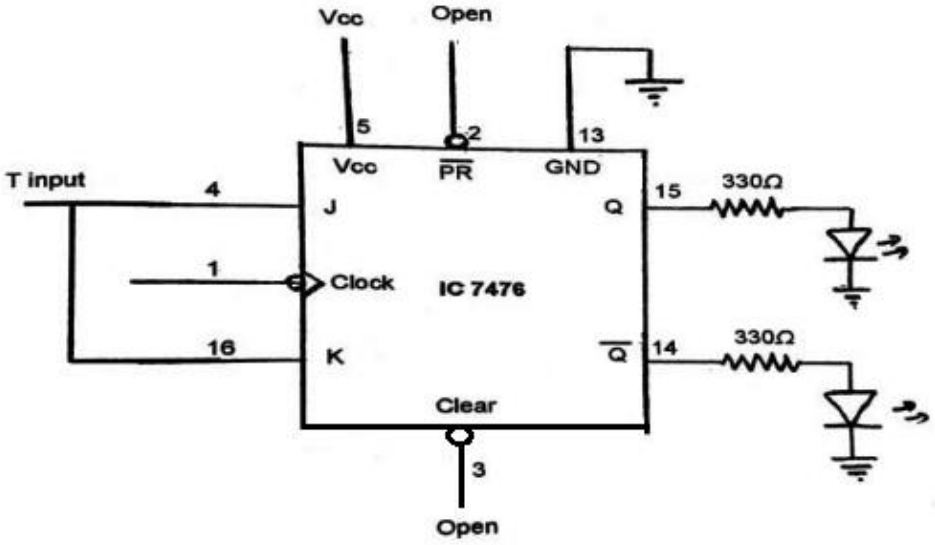


Figure 11.7 Circuit diagram of T flip-flop using IC 7476

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	7476	1
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Study the pin diagram of IC 7476 carefully.
6. Make the connections to convert JK flip flop to T flip flop as per the circuit diagram given in Figure 11.7.
7. Apply logic input to T input and verify the truth table of T flip-flop as per the theoretical output.
8. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Verification of the truth table of T Flip-flop using IC 7476

Sr. No.	$\overline{PR}$	$\overline{CR}$	T	Clk	Theoretical O/P		Practical O/P	
					Q_{n+1}	$\overline{Q}_{n+1}$	Q_{n+1}	$\overline{Q}_{n+1}$
1.	0	1	X	X	1	0		
2.	1	0	X	X	0	1		
3.	1	1	0	X	Previous output			
4.	1	1	X	0	Previous output			
5.	1	1	0	↓	Q_n (Previous O/p)	$\overline{Q}_n$		
6.	1	1	1	↓	$\overline{Q}_n$ (Toggle state)	Qn		

*where Q_{n+1} & $\overline{Q}_{n+1}$ is the next state output

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Can T flip-flop be used in shift registers or counters? Justify your answer.
2. Give two applications of T flip flop.
3. Draw and explain how the T flip-flop can be used as a frequency divider.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References-

1. https://www.alldatasheet.com/datasheet-pdf/pdf/50913/FAIRCHILD/7476.html
2. https://www.futurlec.com/74/IC7476.shtml
3. https://www.electronics-tutorials.ws/sequential/seq_2.html
4. https://nptel.ac.in/courses/108105132
5. https://drive.google.com/file/d/1YYEvVpB0-6xcmaJT1_eO9m8jacFBjTfW/view

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.12: Implementation of 4 bit ripple counter using 7476.

I Practical Significance

In digital logic and computing, a counter is a device, which stores the number of times a particular event or process has occurred in relation to a clock signal. Counter is a sequential circuit used for counting the number of clock pulses. Each pulse applied to the clock input increments or decrements the number in the counter. It is a group of Flip-Flops with a clock signal applied to it. Number of flip-flops connected in cascade. A counter has a natural count of 2^n where "n" is the number of flip-flops in the counter. Hence, a 4-bit counter has 16 output states.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Build simple combinational and sequential circuits.

IV Laboratory Learning Outcome

Implement 4 bit ripple counter using 7476.

V Relevant Affective Domain related outcomes

1. Follow safe practices
2. Demonstrate working as a leader or a team member.
3. Maintain tools and equipment.
4. Follow ethical practices.

VI Relevant Theoretical Background

A ripple counter is an asynchronous counter where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop. Asynchronous counters are also called ripple-counters because of the way the clock pulse ripples away through the flip-flops. When the decimal equivalent of the counter output increases as it receives the clock pulses, then that counter is known as Up Counter. When the decimal equivalent of the counter output decreases as it receives the clock pulses, then that counter is known as Down Counter.

Figure 12.3 shows 4-bit asynchronous (ripple) up-counter designed by using JK flip-flop. In this circuit J and K inputs are to be connected to Vcc (Logic 1 or Logic High) and uncomplemented output Q of the preceding flip-flop is connected to the clock input of the next flip-flop. A 4-bit ripple up-counter will increase the count from 0 to 15. Output Q of the JK flip-flop toggles and changes its state at each negative edge of the clock input as per the truth table shown in figure 12.2.

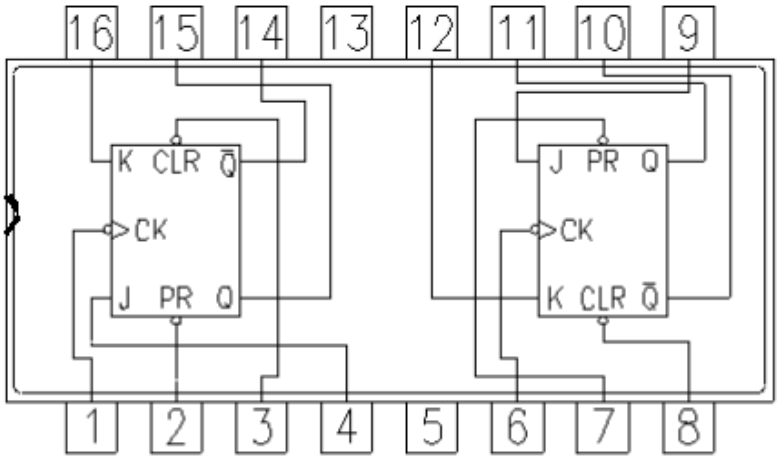


Figure 12.1 Pin diagram of IC-7476 Dual Master-slave JK flip-flop with Preset and Clear inputs

Sr. No.	$\overline{PR}$	$\overline{CR}$	Jn	Kn	Clk	Q_n	$\overline{Q}_n$
1.	0	1	X	X	X	1	0
2.	1	0	X	X	X	0	1
3.	1	1	0	0	X	Previous Output	
4.	1	1	X	X	0	Previous Output	
5.	1	1	1	0	↓	1	0
6.	1	1	0	1	↓	0	1
7.	1	1	1	1	↓	Toggle	

Figure 12.2 Truth Table of JK flip-flop

VII Actual Circuit diagram used in a laboratory with related equipment rating-

A) Suggestive Circuit diagram-

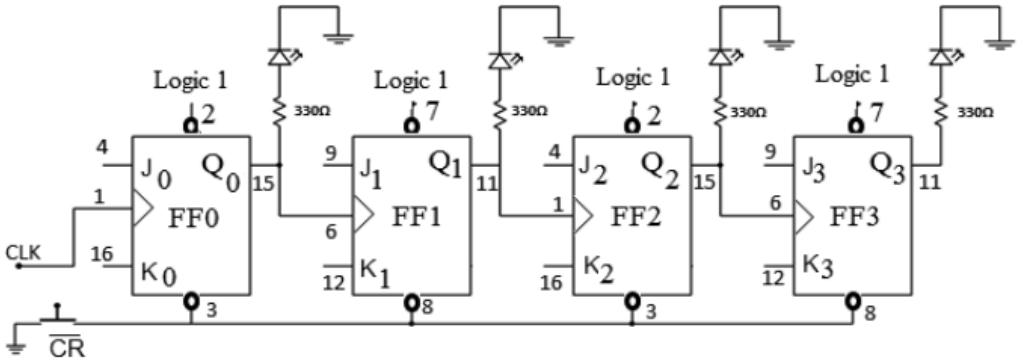


Figure 12.3 Circuit diagram of 4-bit ripple counter using JK flip-flops (IC 7476)

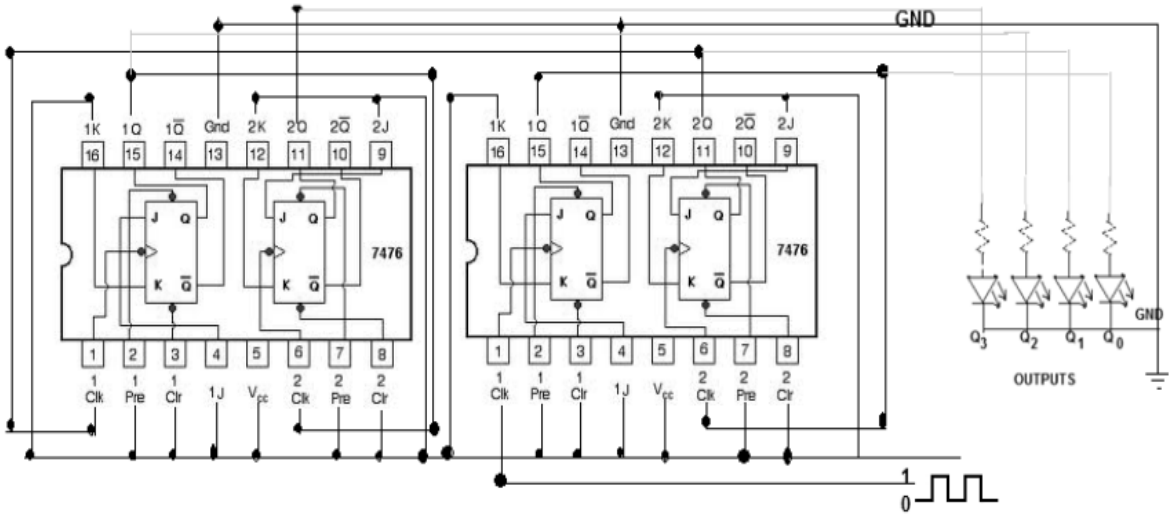


Figure 12.4 Circuit diagram (connection diagram) of 4-bit ripple counter using IC 7476

B) Actual Setup diagram-

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 and ½ digit display with R, V, I measurements, Diode and BJT testing.	1
2	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.	1
3	DC Regulated power supply	Floating DC Supply Voltages 0 -30V; 0-2 A. Automatic Overload (Current Protection). Constant Voltage and Constant Current Operation. Digital Display for Voltage and Current Adjustable Current Limiter. Excellent Line and Load Regulation.	1
4	Breadboard	5.5 cm X 17 cm.	1
5	IC	7476	2
6	LED	Red/Yellow color 5 mm.	2
7	Connecting wires	Single strand 0.6 mm Teflon coating.	As required
8	Resistor	330 Ω or 1KΩ of 0.5W or 0.25W power rating.	2

IX Precautions to be followed

1. Ensure proper earthing to the equipment.
2. Ensure the power switch is in 'off' condition initially.
3. Ensure proper settings of the power supply before use.
4. Switch off the power after completion of the practical.

X Procedure

1. Identify pin configuration of the ICs and test the ICs on the IC tester.
2. If the IC is faulty then keep it in the proper e-waste bin.
3. If the IC is in OK condition then mount it on the breadboard or the trainer kit.
4. Connect pin 14 to Vcc = +5V and pin 7 to ground = 0V of the IC used.
5. Study the pin diagram of IC 7476 carefully.
6. Make the connections as per the circuit diagram of the 4 bit ripple counter given in Figure 12.4.
7. Apply clock pulse to the clock input of JK flip flop (IC 7476) and verify the truth table of 4-bit ripple up-counter as per the theoretical output.
8. Write the reading in each observation table.

XI Resources used

Sr. No.	Name of Resource	Specification	Quantity
1			
2			
3			
4			
5			
6			
7			
8			

XII Actual Procedure

……..

……

..……

……..

……..

……..

……..

……..

XIII Observation Table

Table 1: Truth table of 4-bit Asynchronous (ripple) counter using IC 7476

Sr.No.	Input	Output				
	No. of clock pulses	Q_3	Q_2	Q_1	Q_0	Decimal Equivalent

XIV Result(s)

……

……

……

……

XV Interpretation of results

……

……

……

……

XVI Conclusion and recommendation

……

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identifying CO.

1. Test and write the function of Preset and Clear inputs of JK flip-flop.
2. Give two applications of ripple counters.
3. State the type of flip-flop that can be used in place of JK flip-flop to design a ripple counter.
4. Draw and explain a state diagram of a 4-bit ripple counter.
5. Draw circuit diagram of 4-bit synchronous counter using JK flip-flop.
6. Refer data sheets of different ICs for the application of ripple counter. Suggest IC numbers that are suitable to operate for input voltage of 5V; temperature range is of 30 degree Celsius and 30 MHz of clock frequency.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References-

1. https://www.alldatasheet.com/datasheet-pdf/pdf/50913/FAIRCHILD/7476.html
2. https://www.futurlec.com/74/IC7476.shtml
3. https://nptel.ac.in/courses/108105132
4. https://drive.google.com/file/d/1hduygINHF4OACUnwUKUzxvuwKLONLpjh/view

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related : 15 Marks		**60 %**
1	Selection of ICs and power supply	10%
2	Testing of ICs	20%
3	Proper connections	20%
4	Working in teams	10%
Product Related: 10 Marks		**40%**
5	Result	10%
6	Interpretation of result	05%
7	Conclusion	05%
8	Practical related questions	15%
9	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.13: Assembly language program (ALP) to perform addition of 8-bit data using various addressing modes.

I Practical Significance

The 8051 microcontroller perform arithmetic operations on data. Applications such as BCD, ASCII conversions and checksum byte testing require arithmetic operations. This practical will help the students to develop skills to write assembly program for arithmetic operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program (ALP) to perform addition of 8-bit data.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Microcontroller is a single chip microcomputer made through VLSI fabrication. 8051 is the microcontroller of the MCS-51 family introduced by Intel Corporation. It has inbuilt components such as CPU, internal RAM and ROM, timers/counters, serial ports, interrupts and I/O ports.

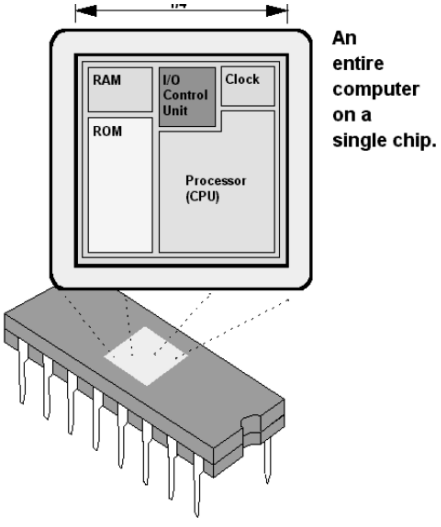


Fig 13.1 System on Chip – 8051

Accumulator is an 8-bit register and is used for all arithmetic and logic operations. To perform arithmetic operations, it is necessary that one of the two operands should be in accumulator. There are four register banks available in 8051; each register bank consists of 8 registers (R0 to R7).

Addressing Modes: A microcontroller provides various methods for accessing data needed in the execution of an instruction. The various methods of accessing the data are called addressing modes.

1. Immediate addressing mode	MOV A,#25H
2. Register addressing mode	MOV A,R0
3. Direct addressing mode	MOV A,40H
4. Register Indirect addressing mode	MOV A,@R0
5. Indexed Addressing mode	MOVC A,@A+DPTR

For addition - ADD A, Source ; A = A + Source

Keil IDE:

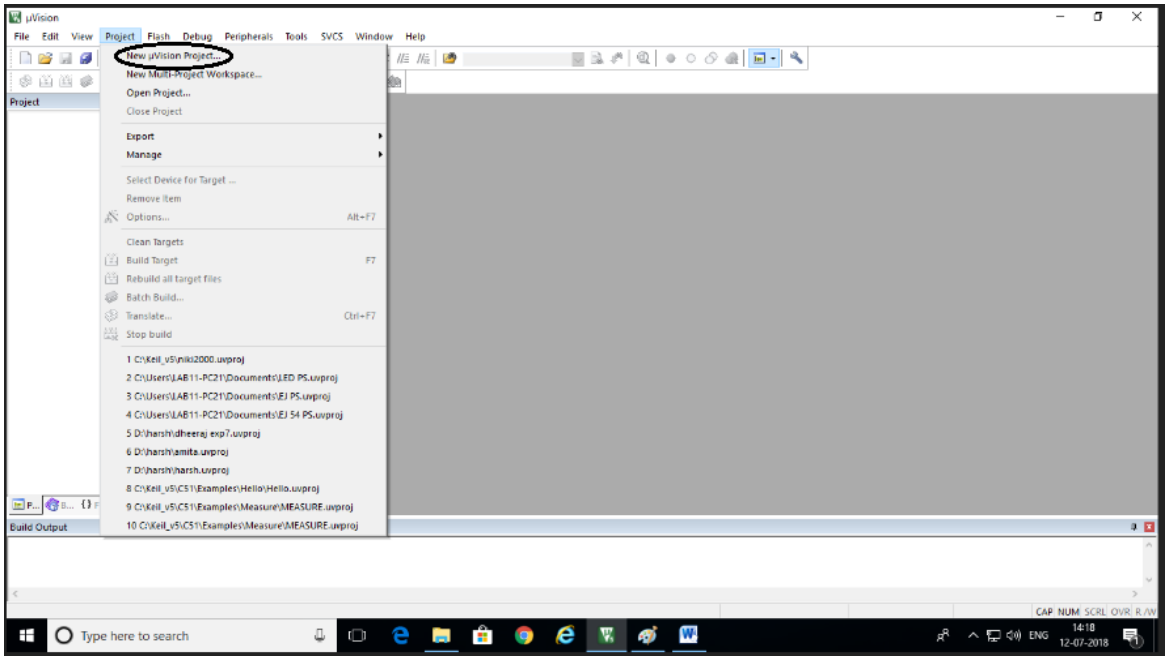
Keil is 8051 development tool which includes a text Editor, Assembler Debugger, linker, Simulator, C-complier, hex converter, locator and some in-built features like logic analyzer to observe various waveforms. It also includes terminal emulator. Keil supports all 8051 derivatives and valuable tool for embedded software development.

Steps for creating a project using Keil software:

1. Start Keil by double clicking on Keil icon. (Keil automatically opens the last project which was opened previously, when Keil was closed).

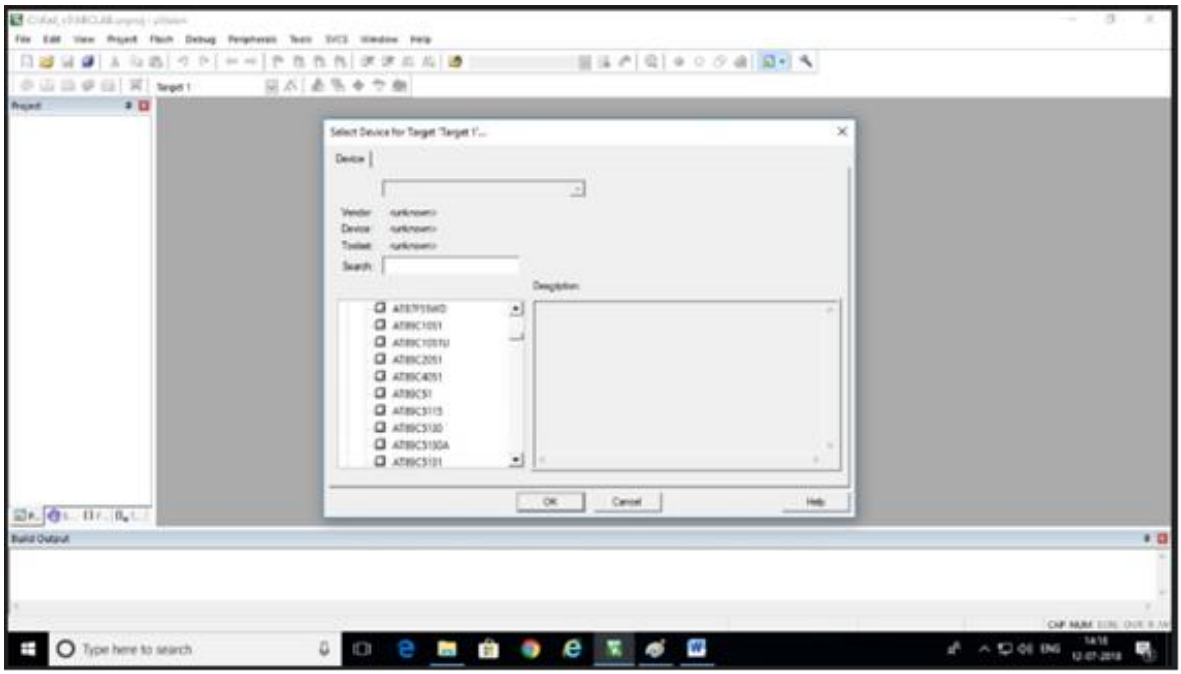
2. To create new project, Click on Project and select new project.

3. Select appropriate location for new project and type project name, click on save button.

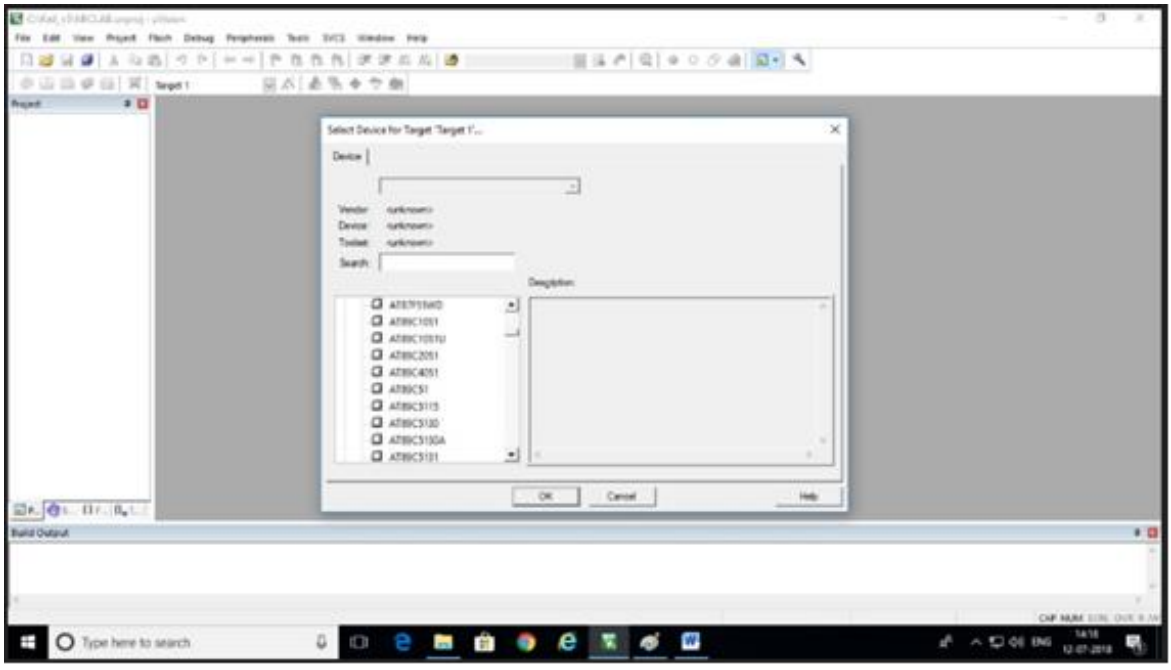


4. "Select device for Target Target-1" window will open. It displays a list of manufacturers of microcontrollers.

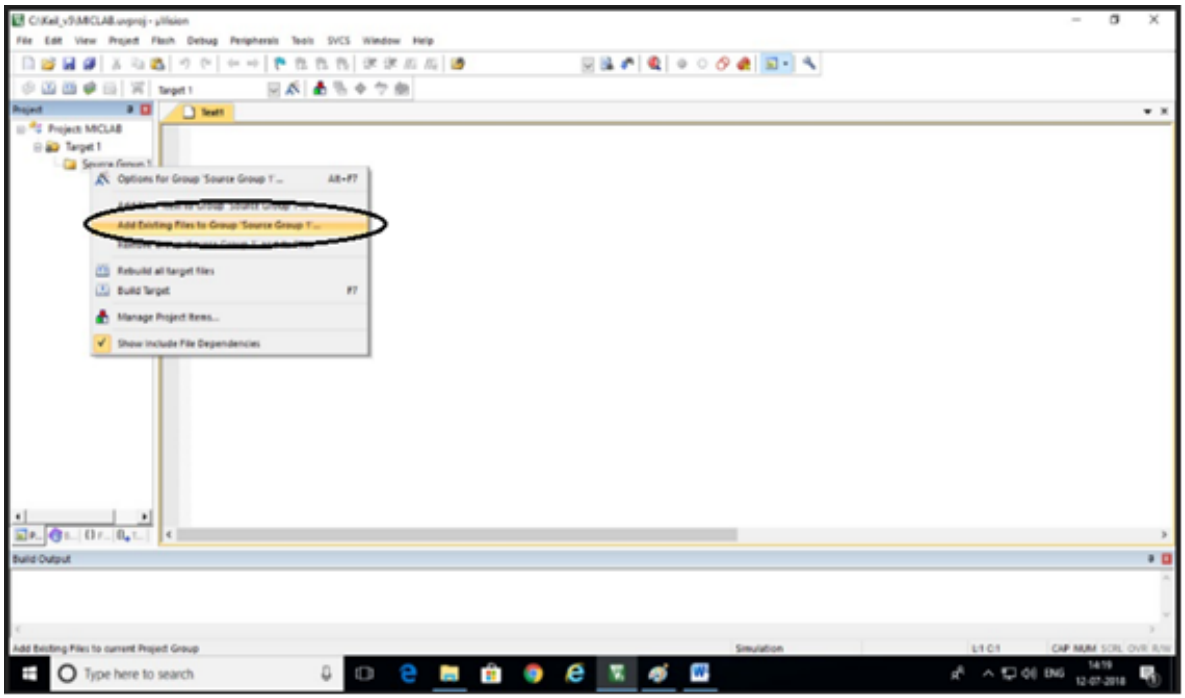
5. Double click on ATMEL or INTEL, list of supported microcontrollers gets displayed. Select 80C51AH from INTEL or AT89C51 (or as per the target board) for ATMEL then click ok.



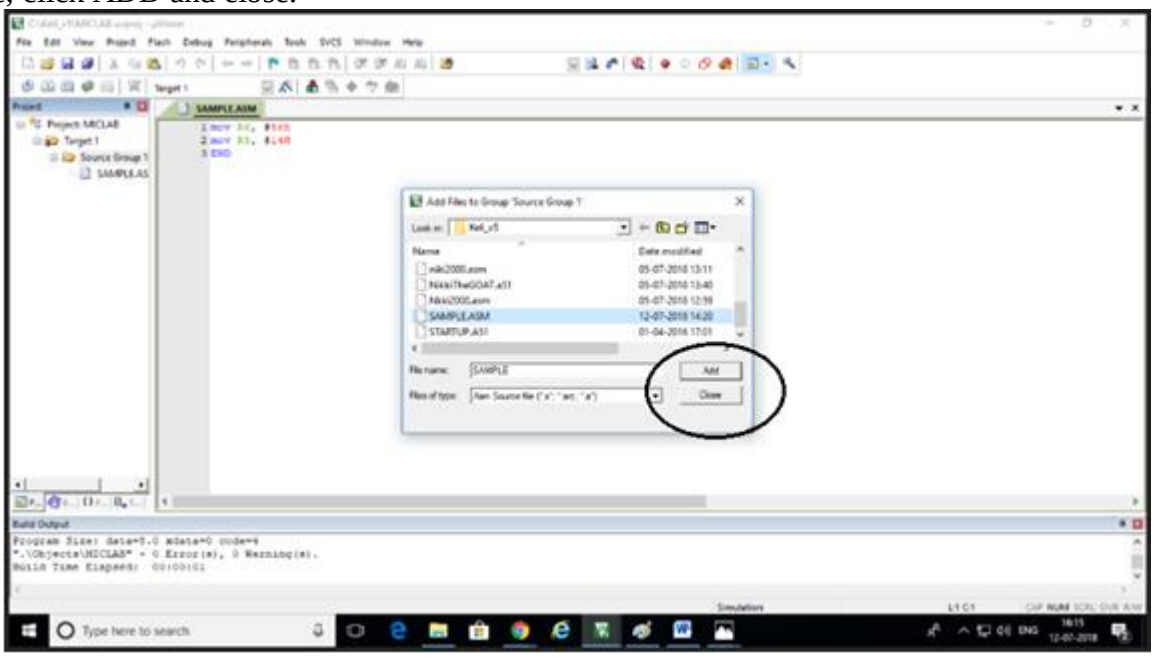
6. Click file pull down menu. Select new, a text editor window will open. Save this file in a same folder where project was stored. Give extension as .ASM or. A51.



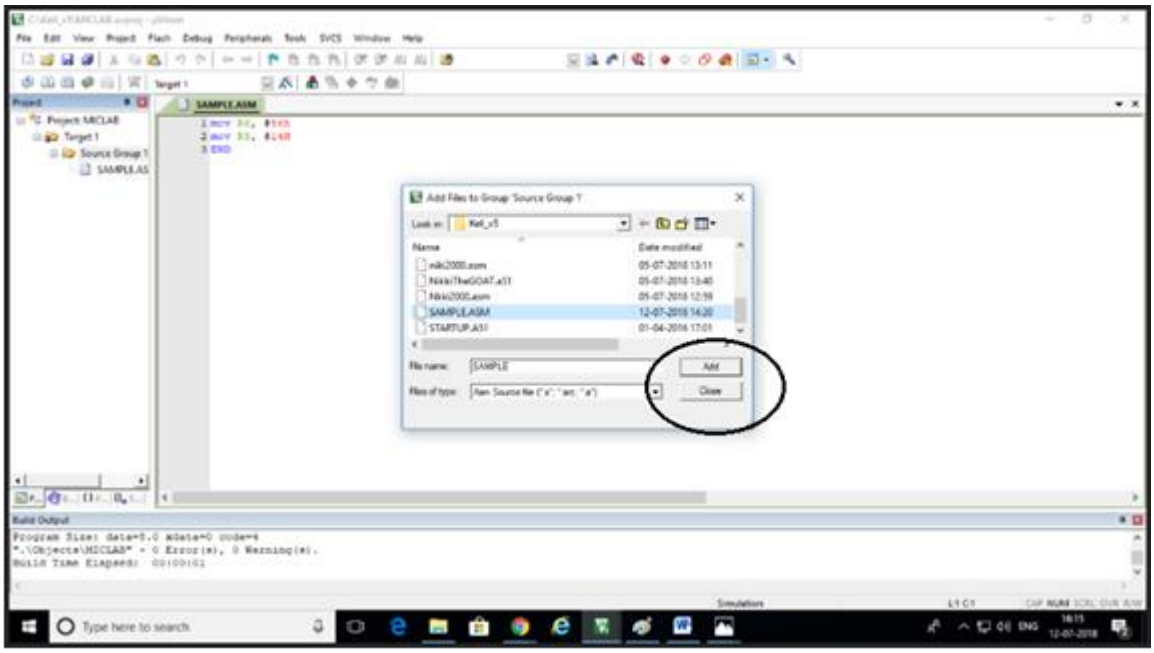
7. On left hand project work space window will display Target1 and Source group1. Right click on source group; Add files to source group 1.



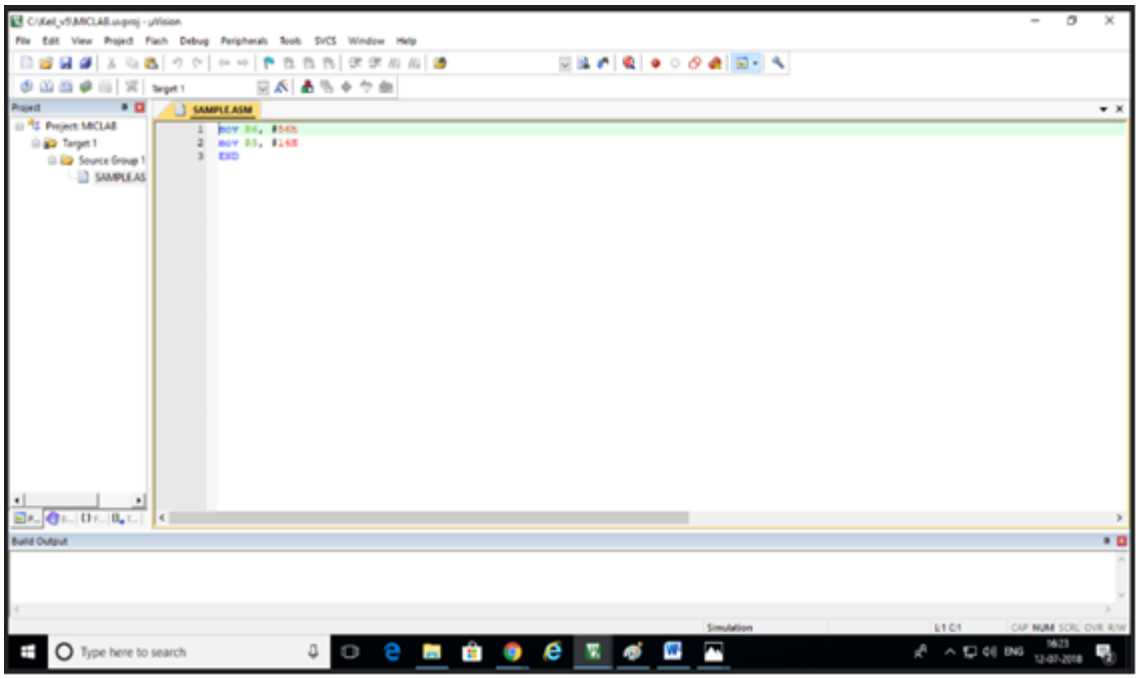
8. Select file type as asm source file. Now all .asm file Name will be displayed. Select appropriate file, click ADD and close.



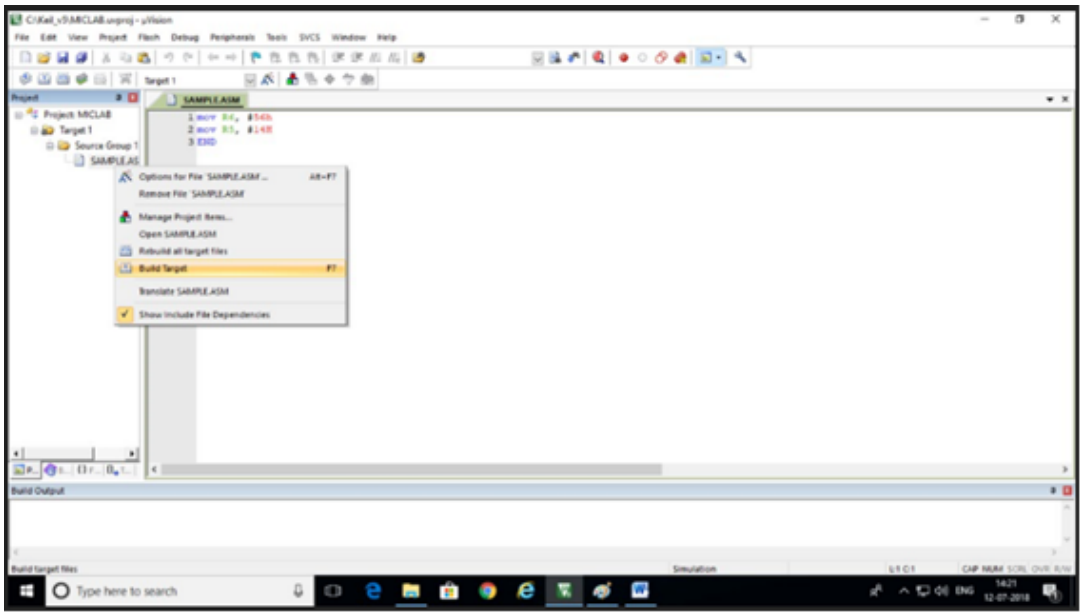
9. Project work space window will display ‘Target 1’ and ‘Source group 1’ with added file name.



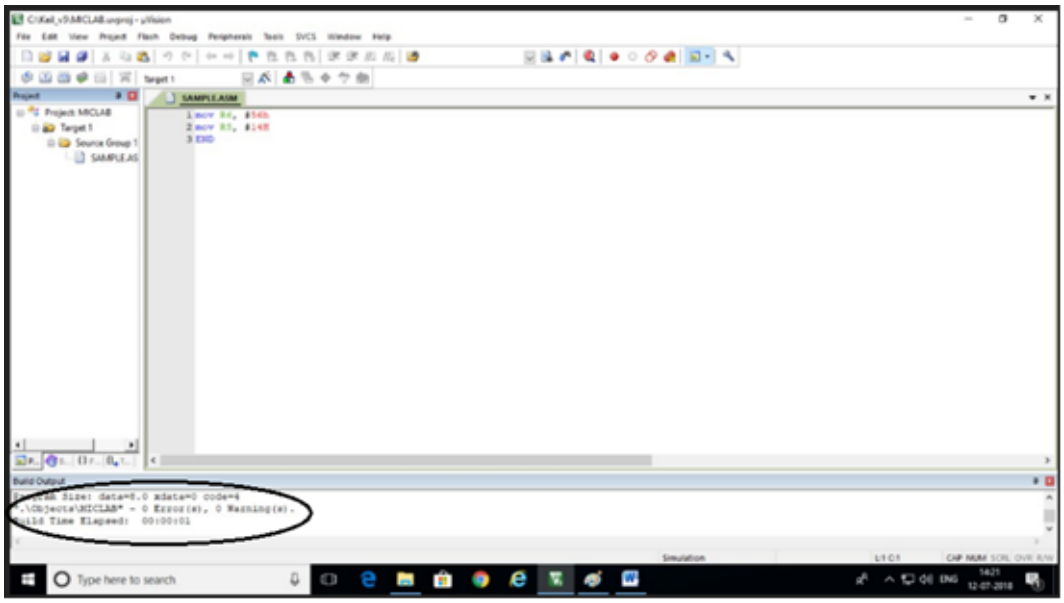
10. Type assembly language program. End with END directive. Save the file periodically.



11. Right click on source group, click on Build target or press F7.



12. Output window will display the errors if any. If there are some errors, then remove the errors and repeat from step number 12 until no errors.

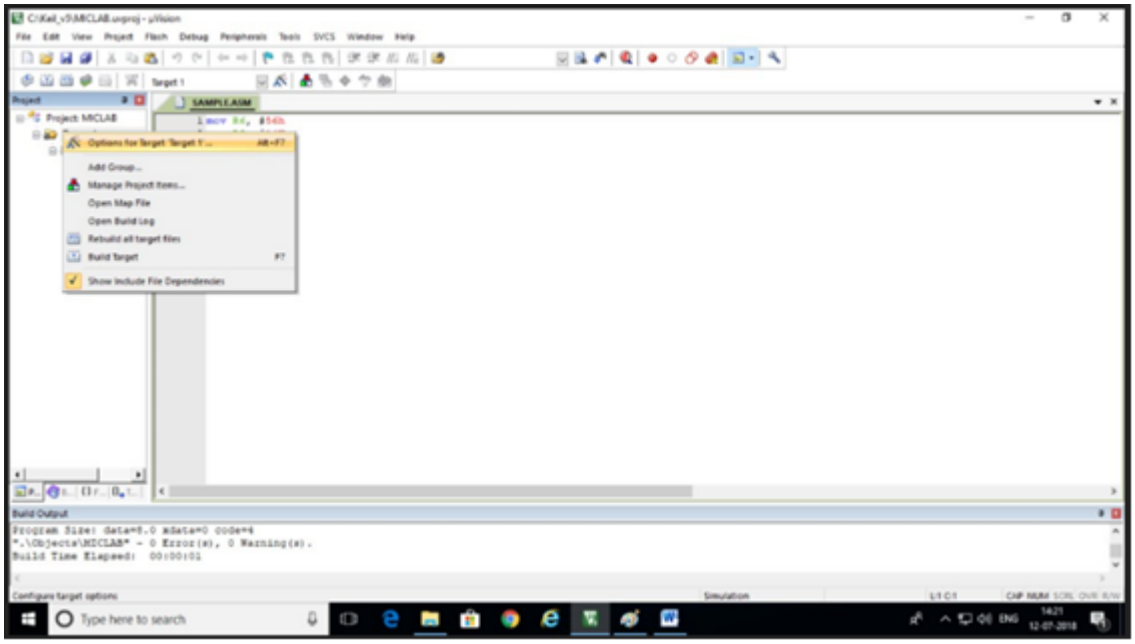


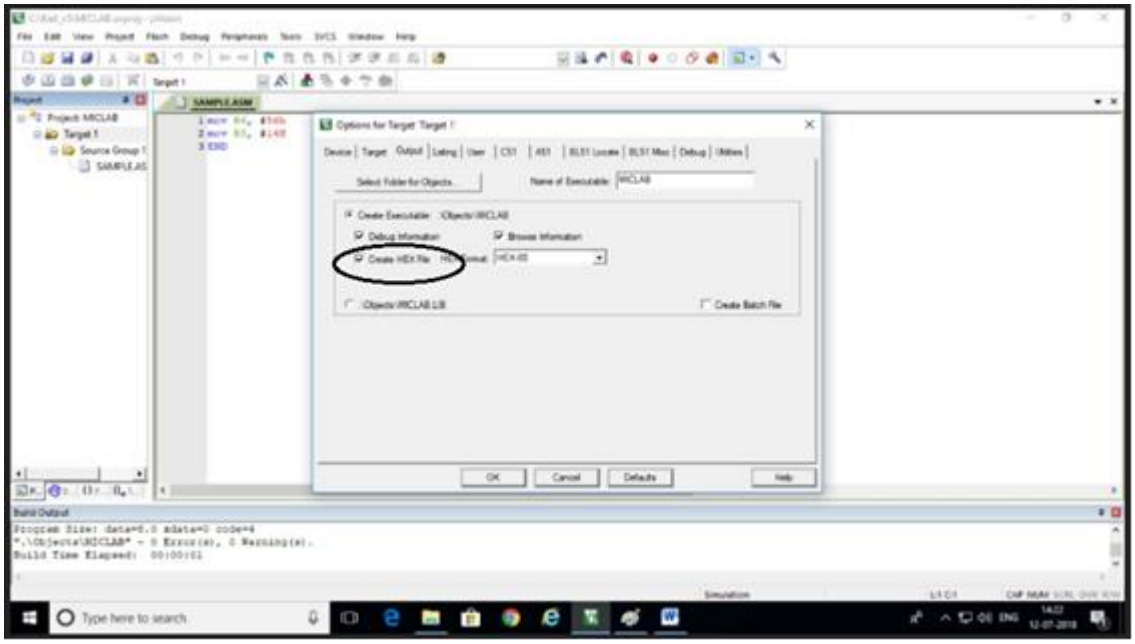
To create a hex file, follow this procedure

a. Right click on target in project window.
b. Click on options for target ‘target 1’.
c. Set target frequency 11.0592 MHz
d. Click on output tab and checkmark the option “create hex file”.
e. Click ok
f. Repeat step 12 again.

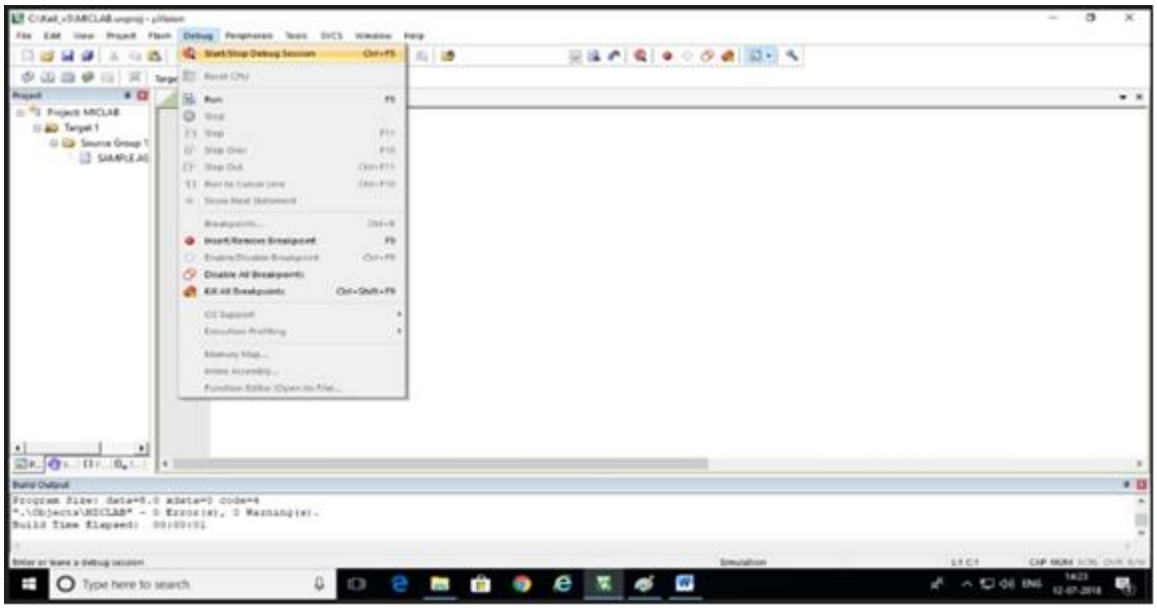
Observe output window. Hex file is created.

This step is optional for the experiments which need only simulation method to observe the results.

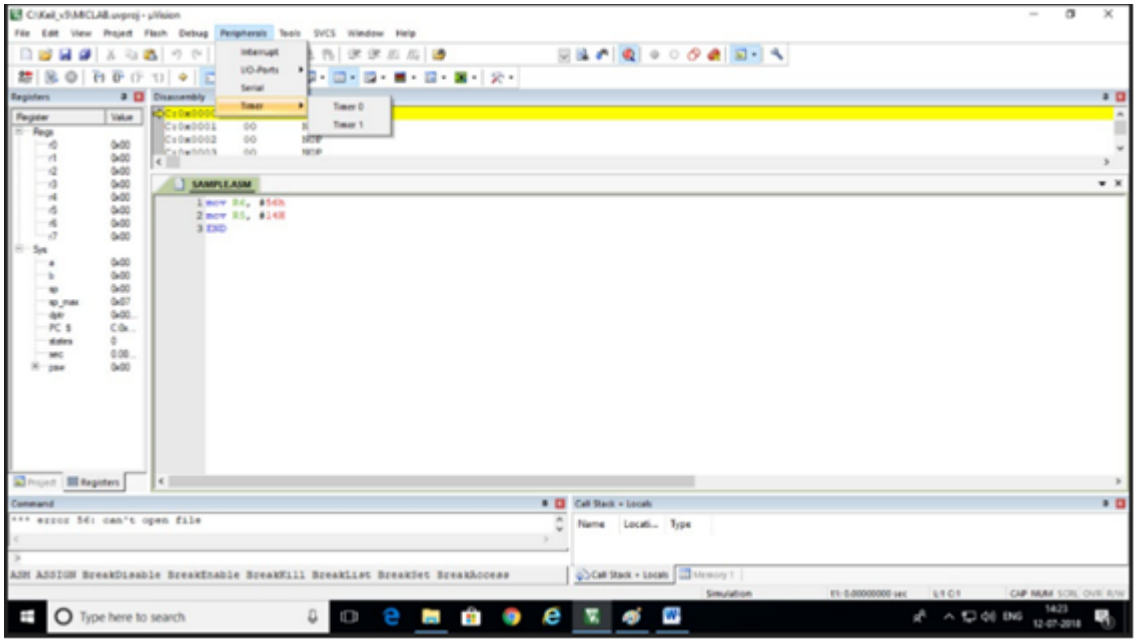




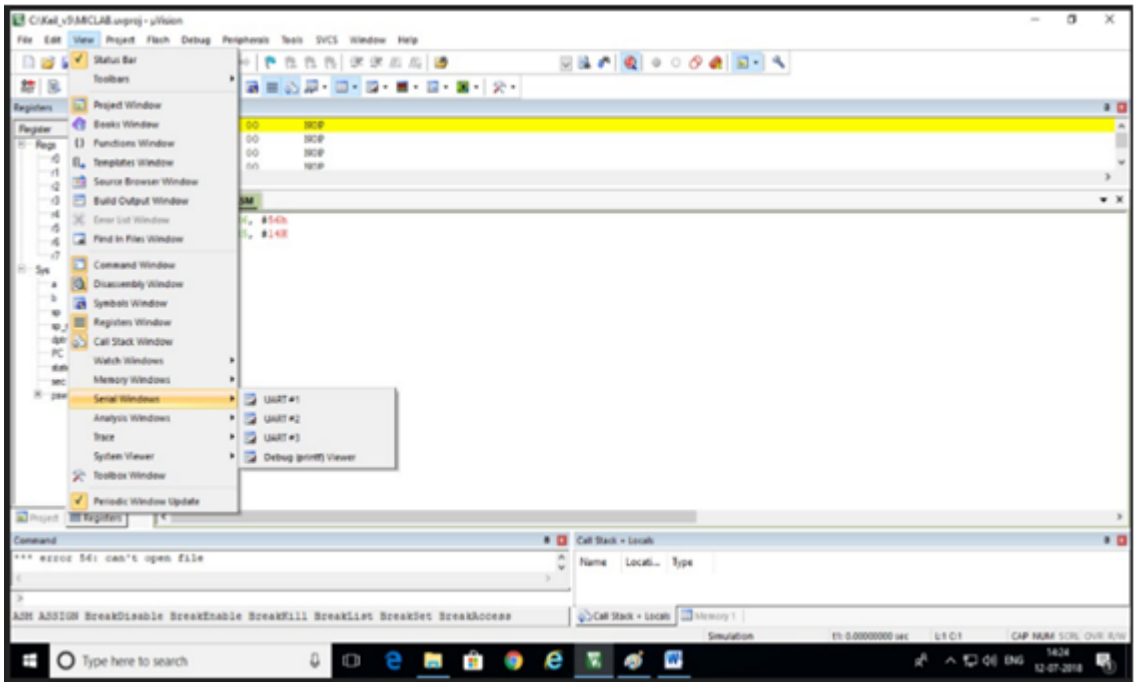
To start the simulation. Click on Debug pull down. Then select start/Stop debug session



On start of debug session, project window will display all internal registers of 8051 and their contents.
To execute the program step by step, go no clicking on “step over” button.
Observe the logic levels of port pins, timers, interrupt etc, by clicking on Peripherals and select appropriate.
Execute the program step by step and observe the logic levels on port pins.



Observe the serial communication by clicking VIEW pull down and select serial window-1 Option.



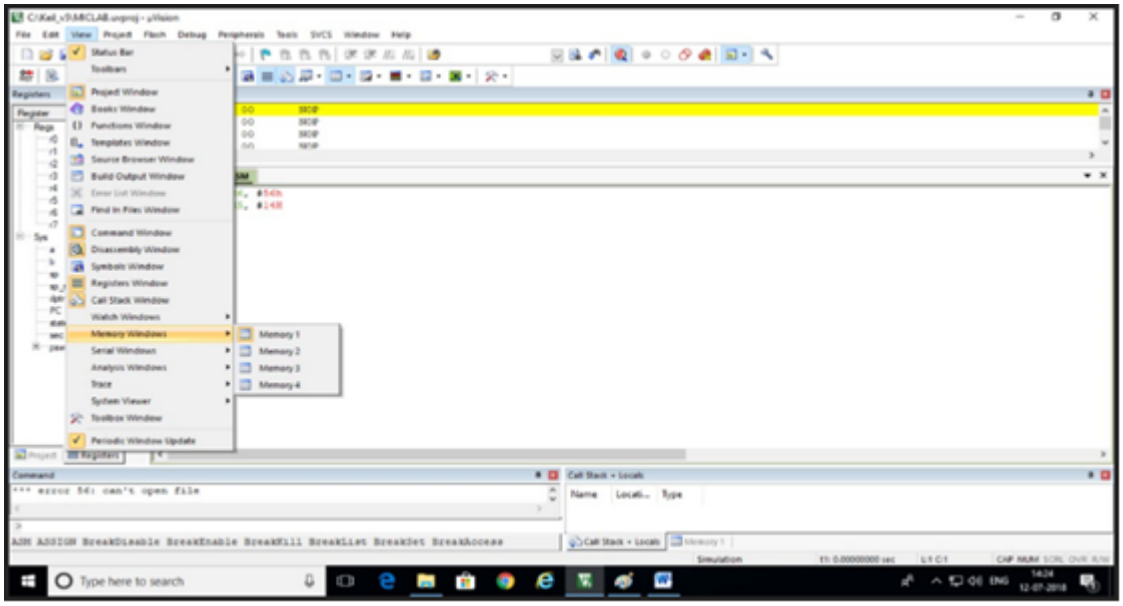
Observe the contents of internal, external and code memory contents.

a. Click on memory window button

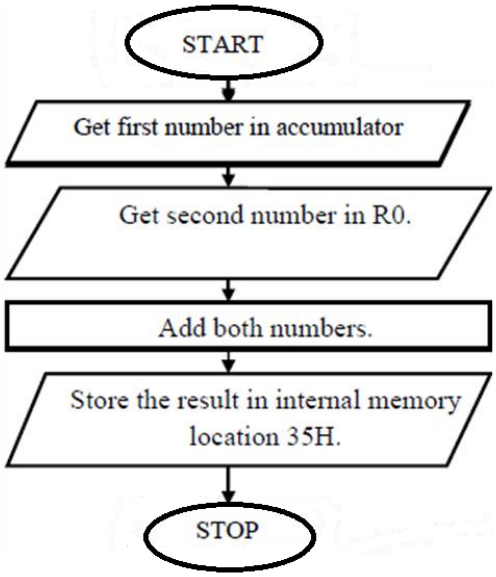
b. Memory window will get displayed near output window with address bar.

c. Type “i: address 8-bit H” for internal memory.” X: address16 bit H” for external memory for code memory type “C: address”.

d. To modify the contents of memory, right click on contents of any memory location and enter new valued to be loaded in that memory location



Addition

Sample program: Write and execute a program to add data 40H and 30H and store result in 35H internal memory	**Student activity:** **Write a program to add two 8-bit numbers. Assume numbers are stored in the Internal memory locations 30H and 31H store result at 32H.**
Algorithm: 1. Get first number in accumulator 2. Get second number in R0 3. Add both numbers 4. Store the result in internal memory location 35H 5. Stop	**Algorithm:**
Flowchart:  	**Flowchart**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV A,#30H	Read 1st number into A register		
MOV R0,#40H	Read 2nd number into R0 register		
ADD A,R0	Add content of A and R0 register		
MOV 35H,A	Store result at 35H internal memory		
END	End of the program		

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Sr. No.	Data stored in Register/ Memory for the operation	Result after execution	Status of flags
1			
2			
3			

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write the result of addition of two numbers 88H and 4AH.
2. Write assembly language program to perform addition of two 16-bit numbers.

3. List any four instructions for addition operation and write meaning of each.
4. Write assembly program to perform series addition of 10 numbers.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51
4. https://electronicsforyou.in/8051-program-for-addition-of-two-16-bit-numbers/

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.14: Assembly language program (ALP) to perform subtraction of 8-bit data using various addressing modes.

I Practical Significance

The 8051 microcontroller perform arithmetic operations on data. Applications such as BCD, ASCII conversions and checksum byte testing require arithmetic operations. This practical will help the students to develop skills to write assembly program for arithmetic operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program (ALP) to perform subtraction of 8-bit data.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Accumulator is an 8-bit register and is used for all arithmetic and logic operations. To perform arithmetic operations, it is necessary that one of the two operands should be in accumulator. There are four register banks available in 8051; each register bank consists of 8 registers (R0 to R7).

For subtraction - SUBB A, Source ; A = A – Source – CY

Subtraction can be done by taking the two`s complement of the number to be subtracted (subtrahend), and adding it to another number (minuend). Register A is the destination address for the subtraction. All addressing modes may be used for source address.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

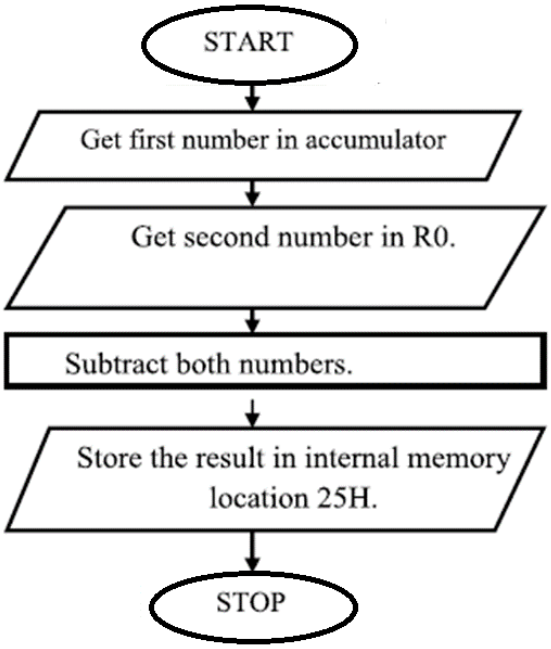
6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

Subtraction

Sample program: Write and execute a program to subtract data 50H and 20H and store result in 25H internal memory	**Student activity:** **Write a program to subtract two 8-bit numbers. Assume numbers are stored in the Internal memory locations 40H and 41H store result at 42H.**
Algorithm: 1. Get first number in accumulator 2. Get second number in R0 3. Perform subtraction of two numbers 4. Store the result in internal memory location 25H 5. Stop	**Algorithm:**

Flowchart:  	**Flowchart**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
CLR C	Clear carry flag		
MOV A,#50H	Read 1st number into A register		
MOV R0,#20H	Read 2nd number into R0 register		
SUBB A,R0	Add content of A and R0 register		
MOV 25H,A	Store result at 35H internal memory		
END	End of the program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Sr. No.	Data stored in Register/ Memory for the operation	Result after execution	Status of flags
1			
2			
3			

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write assembly program to perform subtraction of two 16-bit numbers.
2. If SUBB A, R4 is executed, then actually what operation is being applied?
3. Write significance of carry flag while doing subtraction.

4. On power up, the 8051 uses which RAM locations for register R0- R7.

[Space for Answers]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.15: Assembly language program (ALP) to perform multiplication of 8-bit data, take the input data from port1 and display the output data on port 2.

I Practical Significance

The 8051 microcontroller perform arithmetic operations on data. Applications such as BCD, ASCII conversions and checksum byte testing require arithmetic operations. Arithmetic operators are used to perform calculations on numeric data in a field. This practical will help the students to develop skills to write assembly program for arithmetic operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program (ALP) to perform multiplication of 8-bit data.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Accumulator is an 8-bit register and is used for all arithmetic and logic operations. To perform arithmetic operations, it is necessary that one of the two operands should be in accumulator. There are four register banks available in 8051; each register bank consists of 8 registers (R0 to R7).

For multiplication - MUL AB ; A (LSB Result) = Multiply A x B
; B (MSB Result)

The register A and B will be used for multiplication. No other registers can be used for multiplication. The result of the multiplication may exceed the 8-bit size. So, the higher order byte is stored at register B, and lower order byte will be in the Accumulator A after multiplication.

8051 microcontrollers have 4 I/O ports i.e. port 0, port 1, port 2 and port 3 each of 8-bit, which can be configured as input or output. Hence, total 32 input/output pins allow the microcontroller to be connected with the peripheral devices.

Pin configuration: The pin can be configured as 1 for input and 0 for output as per the logic state.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

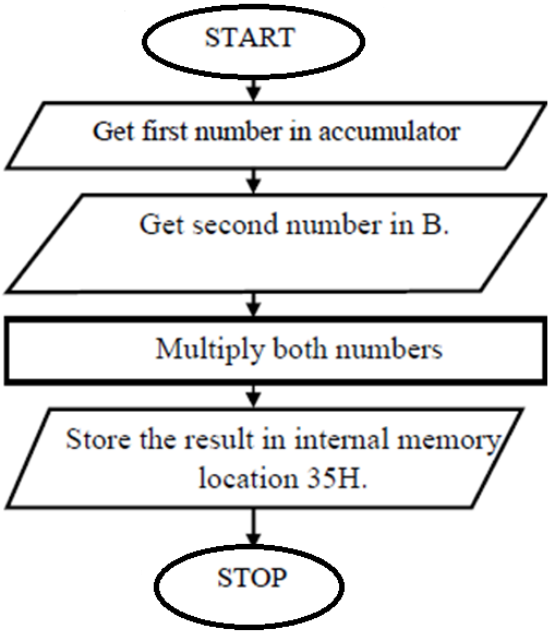
6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

Multiplication

Sample program: Write and execute a program to multiply data 04H and 02H and store result in 35H internal memory	**Student activity:** **Write a program to multiply two 8-bit numbers. Assume numbers are stored in the Internal memory locations 40H and 41H store LSB_result at 42H and MSB_result at 43H.**
Algorithm: 1. Get first number in accumulator 2. Get second number in B. 3. Multiply both numbers 4. Store the LSB result in internal memory location 35H. 5. Store the LSB result in internal memory location 35H. 6. Stop	**Algorithm:**

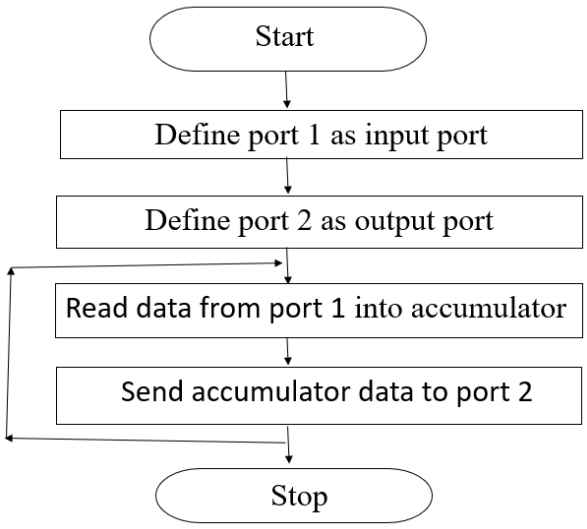
Flowchart:	**Flowchart**
	

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV A,#04H	Read 1st number into A register		
MOV B,#02H	Read 2nd number into B register		
MUL AB	Multiply content of A and B register		
MOV 35H,A	Store LSB result at 35H internal memory		
MOV 36H,B	Store MSB result at 36H internal memory		
END	End of the program		

Take the input data from port and display the output data on other port.

Sample program 2: Write and execute a program to Take the input data from port1 and display the output data on port 2.	**Student activity: Write a program to take the input data from port1 and display the output data on port 2 and port 3.**
Algorithm: 1. Define port 1 as input port. 2. Define port 2 as output port. 3. Read port 1 data into accumulator 4. Send accumulator data to port 2.	**Algorithm:**

5. Repeat the program from step 3. 6. Stop	
Flowchart:  	**Flowchart**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV P1, #0FFH	Define port 1 as input		
MOV P2, #00H	Define port 2 as output		
UP: MOV A, P1	Read port 1 data into accumulator		
MOV P2, A	Send accumulator data to port 2		
SJMP UP	To repeat procedure, jump to label UP		
END	End of the program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……..

..……

……..

XIII Observation Table

Sr. No.	Data stored in Register/ Memory for the operation	Result after execution	Status of flags
1			
2			
3			

XIV Result(s)

……..

……..

..……

……..

XV Interpretation of results

……..

.....……..

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write assembly program to receive data from port 1, determine whether bit 2 is high, and then send the number FFH to port 3
2. Write length in byte and oscillator period required to execute MUL AB instruction.
3. MUL A, R1 is these a valid 8051 instructions? Explain your answer.
4. Write code to send 55H to ports P1 and P2.

[Space for Answers]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.16: Assembly language program (ALP) to perform division of 8-bit data, take the input data from port 2 and display the output data on port 0.

I Practical Significance

The 8051 microcontroller perform arithmetic operations on data. Applications such as BCD, ASCII conversions and checksum byte testing require arithmetic operations. Arithmetic operators are used to perform calculations on numeric data in a field. This practical will help the students to develop skills to write assembly program for arithmetic operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program (ALP) to perform division of 8-bit data.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Accumulator is an 8-bit register and is used for all arithmetic and logic operations. To perform arithmetic operations, it is necessary that one of the two operands should be in accumulator. There are four register banks available in 8051; each register bank consists of 8 registers (R0 to R7).

```
For division -    DIV AB          ; A (Quotient Result) = Divide A by B
                                  ; B (Remainder Result)
```

The register A and B will be used for division. No other registers can be used for division. The numerator must be in register A and the denominator must be in B. After division Quotient is stored in A register and Remainder in B.

8051 microcontrollers have 4 I/O ports i.e. port 0, port 1, port 2 and port 3 each of 8-bit, which can be configured as input or output. Hence, total 32 input/output pins allow the microcontroller to be connected with the peripheral devices.

Pin configuration: The pin can be configured as 1 for input and 0 for output as per the logic state.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

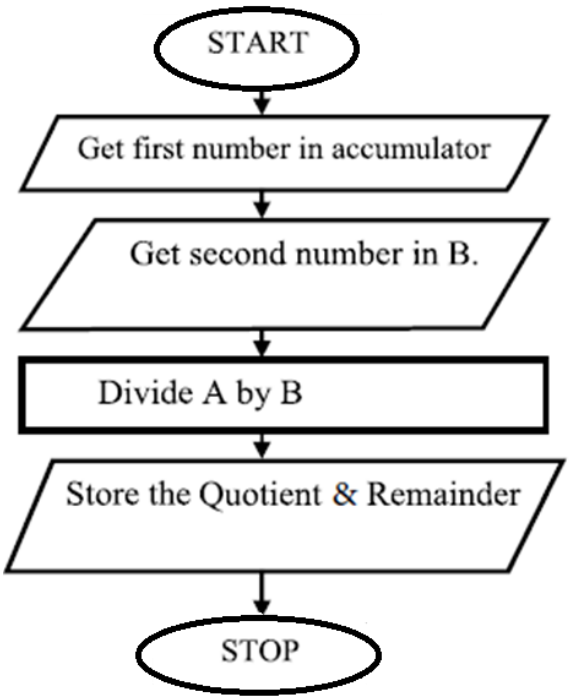
6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

Division

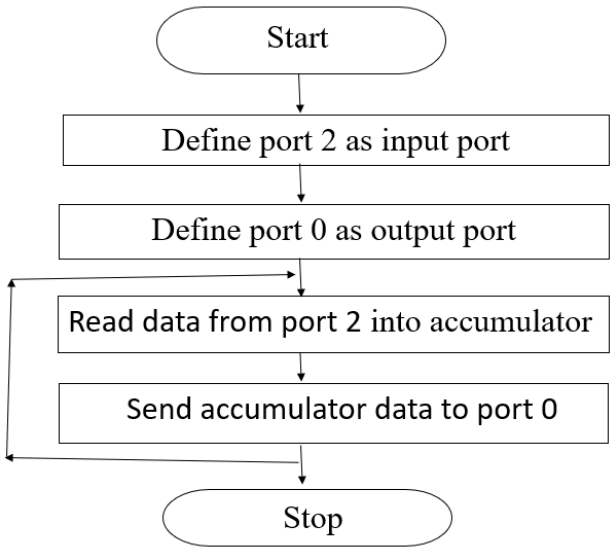
Sample program: Write and execute a program to divide data 09H by 02H and store result in 35H and 36H internal memory	**Student activity:** **Write a program to divide two 8-bit numbers. Assume numbers are stored in the Internal memory locations 40H and 41H store Quotient at 42H and Remainder result at 43H.**
Algorithm: 1. Get first number in accumulator 2. Get second number in B. 3. Divide content of A by content of B 4. Store the result (Quotient) in internal memory location 35H. 5. Store the result (Remainder) in internal memory location 36H. 6. Stop	**Algorithm:**

Flowchart:	**Flowchart**
	

Assembly language program

Instructions	**Comments**	**Instructions**	**Comments**
ORG 0000H	Start of program		
MOV A, #09H	Read 1st number into A register		
MOV B, #02H	Read 2nd number into B register		
DIV AB	Multiply content of A and B register		
MOV 35H, A	Store result (Quotient) at 35H internal memory		
MOV 36H, B	Store result (Remainder) at 36H internal memory		
END	End of the program		

Take the input data from port and display the output data on other port.

Sample program: Write and execute a program to Take the input data from port 2 and display the output data on port 0.	**Student activity: Write a program to take the input data from port 2 and display the output data on port 1 and port 3.**
Algorithm: 1. Define port 2 as input port. 2. Define port 0 as output port. 3. Read port 2 data into accumulator 4. Send accumulator data to port 0. 5. Repeat the program from step 3. 6. Stop	**Algorithm:**
Flowchart:  	**Flowchart:**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV P2, #0FFH	Define port 2 as input		
MOV P0, #00H	Define port 0 as output		
UP: MOV A, P2	Read port 2 data into accumulator		
MOV P0, A	Send accumulator data to port 0		
SJMP UP	To repeat procedure, jump to label UP		
END	End of the program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Sr. No.	Data stored in Register/ Memory for the operation	Result after execution	Status of flags
1			
2			
3			

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write length in byte and oscillator period required to execute DIV AB instruction.
2. DIV A, R1 is these a valid 8051 instructions? Explain your answer.
3. Write assembly code to convert hexadecimal number to decimal.
4. Write assembly code to toggle the bits of ports P3 continuously.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.17: Assembly language program to transfer data from source to destination location of internal data memory.

I Practical Significance

Data transfer is a process of moving or copying information from one location to other location within internal or external memory. To save the results of certain operations, to create lookup tables etc. This practical will help the students to develop skills to write assembly program for memory related data transfer operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program to transfer data using internal data memory.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

The 8051 microcontroller's memory is divided into Program Memory and Data Memory. In the 8051 microcontroller, Internal RAM (data memory) - 128 bytes. Internal memory (code memory) - 4 kB (ROM).

Program Memory (ROM) is used for permanent saving program being executed, while Data Memory (RAM) is used for temporarily storing and keeping intermediate results and variables.

The block transfer is the process of transfer the content of group of memory location from source address to destination address, after block transfer content of destination replace by new contents.

MEMORY WINDOW:
Before execution:
D:0x50H: 22 AB 3D 44 55

D:0X60H: 00 00 00 00 00

After execution:
D:0x50H: 22 AB 3D 44 55

D:0X60H: 22 AB 3D 44 55

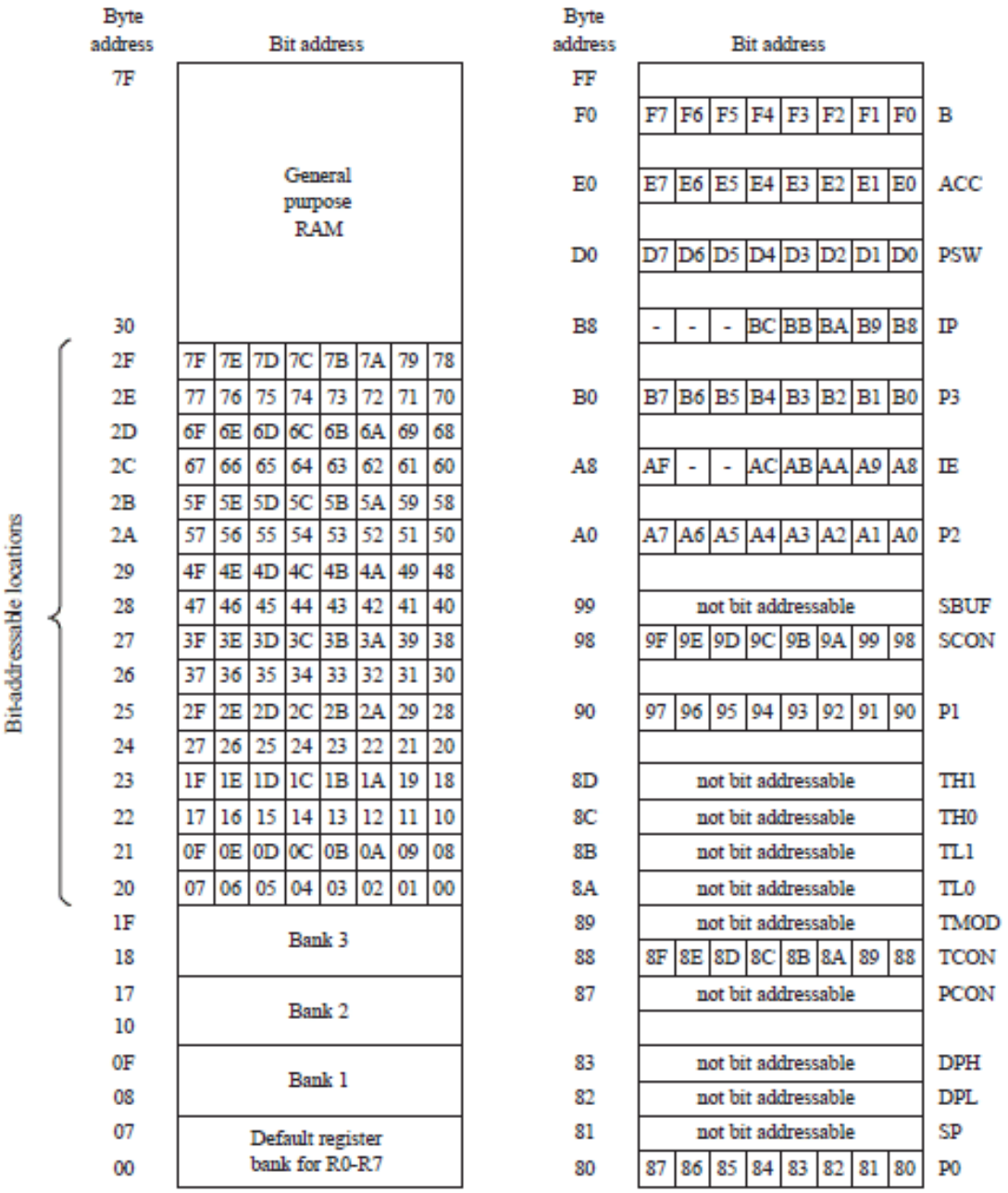


Fig. 17.1 RAM memory of 8051

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 or 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

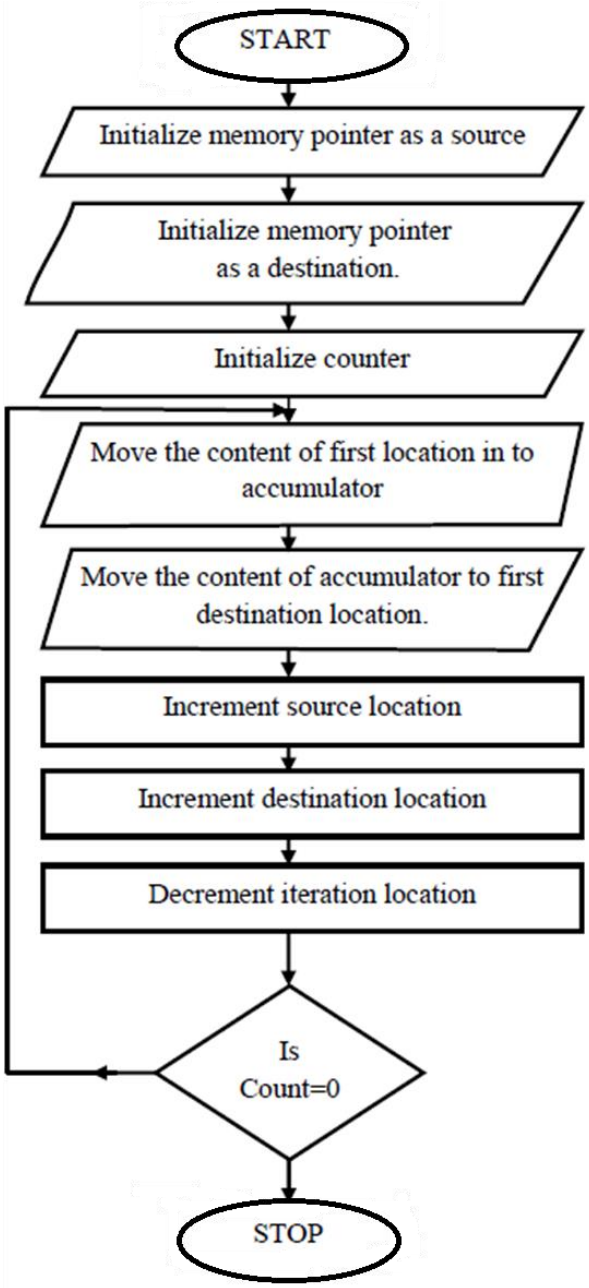
8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

Data transfer

Sample program: Write and execute a program to transfer five data bytes of internal memory location from 50H to 60H onwards.	**Student activity: Write and execute a program to transfer ten data bytes of internal memory location from 20H to 30H onwards.**
Algorithm: 1. Start 2. Initialize memory pointer as a source. 3. Initialize memory pointer as a destination. 4. Initialize counter. 5. Move the content of first location in to accumulator. 6. Move the content of accumulator to first destination location. 7. Increment source location.	**Algorithm:**

8. Increment destination location.
9. Decrement iteration count and if not zero jump to step 5.
10. Stop

Flowchart:



Flowchart

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV R0,#50H	Source memory pointer		
MOV R1,#60H	Destination memory pointer		
MOV R7,#05H	Set R7 register as Counter		
UP:MOV A,@R0	Get the number from source into accumulator		
MOV @R1,A	Move it into destination		
INC R0	Point on next source memory location		
INC R1	Point on next destination memory location		
DJNZ R7,UP	Repeat the transfer till counter R7=0		
END	End of Program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

…….

..……

……..

XIII Observation Table

Before execution						After execution		
	Address	Contents		Address	Contents		Address	Contents
Source Memory			Destination Memory			Destination Memory		

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write instructions to copy the byte in Accumulator to register R2 using at least two different addressing modes.
2. State significance of DJNZ instruction in this program?
3. Write names of I/O ports and pins, which are used to interface external memory to 8051.
4. List 16 bit registers available in 8051 microcontroller series.

[Space for Answers]

..…….....…

..…….....…

..…….....…

..…….....…

..…….....…

..…….....…

..…….....…

..…….....…

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.18: Assembly language program to transfer data from source to destination location of external data memory.

I Practical Significance

The 8051 has a limited amount of internal memory, it is possible to extend the memory capacity by interfacing it with external memory devices. Data transfer is a process of moving or copying information from one location to other location within internal or external data memory. To save the results of certain operations, to create lookup tables etc. This practical will help the students to develop skills to write assembly program for memory related data transfer operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute program in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program to transfer data using external data memory.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

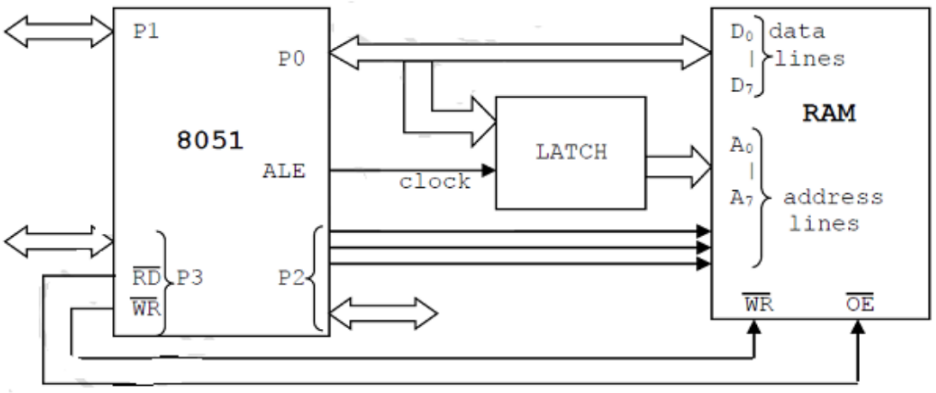


Fig. 18.1 External RAM memory interfacing with 8051

Figure 18.1 shows how to connect or interface external RAM (data memory) to 8051. Port 0 is used as multiplexed data and address lines. Address lines are decoded using external latch and ALE signal from 8051 to provide lower order (A7-A0) address lines. Port 2 gives higher order address lines. RD and WR signals from 8051 select the memory read and memory write operations respectively.
Note: RD and WR signals: generally, P3.6 and P3.7 pins of port 3 are used to generate memory read and memory write signals. Remaining pins of port 3 i.e. P3.0-P3.5 can be used for other functions.

Instructions to Access External Data Memory:

Mnemonic	Operation
MOVX A, @Ri	In this operation, it will copy the contents of the external address in Ri to A.
MOVX A, @DPTR	Copy the contents of the external address pointed by DPTR to A.
MOVX @Ri, A	Copy data from A to the external address in Ri
MOVX DPTR, A	Copy data from A to the external address pointed by DPTR.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

Data transfer external RAM

Sample program: Write and execute a program for block transfer. Take data of external memory location 4000H and transfer to 5000H external memory location.	**Student activity: Write and execute a program for block transfer. Take data of external RAM memory location 6000H and transfer to 7000H external memory location.**
Algorithm: 1. Start 2. Initialize counter. 3. Initialize memory pointer as a source by loading 4000H in DPTR 4. Move the content of first location in to accumulator. 5. Initialize memory pointer as a destination 6. by loading 50H in DPH. 7. Move the content of accumulator to first destination location. 8. Increment DPTR. 9. Decrement iteration count and if not zero jump to step 3. 10. Stop	**Algorithm:**

Assembly language program

Instructions	**Comments**	**Instructions**	**Comments**
ORG 0000H	Start of program		
MOV R0,#05H	Set the counter		
MOV 82H,#00H	DPL = 00H		
UP:MOV 83H,#40H	DPH= 40H		
MOVX A,@DPTR	Get the number from external source memory into accumulator		
MOV 83H,#50H	DPH = 50H		
MOVX @DPTR,A	Get the number from accumulator to destination memory		
INC DPTR	Increment DPTR		
DJNZ R0,UP	Repeat the transfer till counter R0= 0		
END	End of Program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Before execution			After execution		
	Address	Contents		Address	Contents
Source Memory			Destination Memory		

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write instructions to access external ROM / Program Memory.
2. Write instruction to access external RAM / Data memory.
3. Write DPL and DPH memory address.
4. Explain the role of DPTR used in the program.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.19: Assembly language program to exchange data from source to destination memory location.

I Practical Significance

Data transfer is a process of moving or copying information from one location to other location within internal or external data memory. To save the results of certain operations, to create lookup tables etc. This practical will help the students to develop skills to write assembly program for memory related data transfer operations.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute program in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program to exchange data of memory locations.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

The 8051-microcontroller memory is divided into Program Memory and Data Memory. Program Memory (ROM) is used for permanent saving program being executed, while Data Memory (RAM) is used for temporarily storing and keeping intermediate results and variables.

The block exchange is the process of transfer the content of group of memory location from source address to destination address and vice versa, after block exchange content of source and destination memory replace by new contents.

MEMORY WINDOW:
Before execution:
D:0x50H: 01 02 03 04 05

D:0X60H: 06 07 08 09 10

After execution:
D:0x50H: 06 07 08 09 10

D:0X60H: 01 02 03 04 05

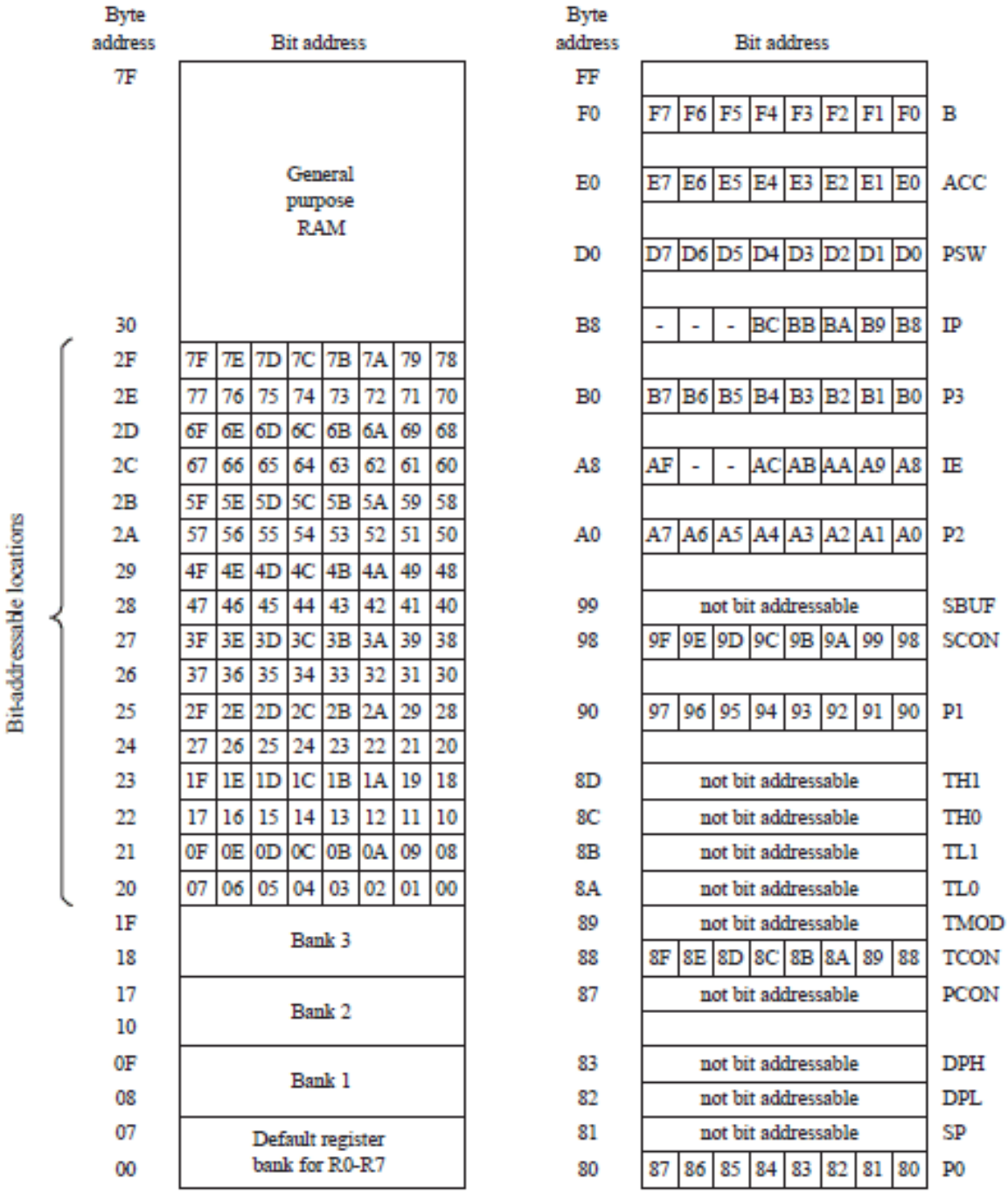


Fig. 19.1 RAM memory of 8051

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers/memory in observation table.

Data exchange

Sample program: Write and execute a program to exchange five data bytes of internal memory location from 50H to 60H onwards.	**Student activity: Write and execute a program to exchange five data bytes of internal memory location from 20H to 30H onwards.**
Algorithm: 1. Start 2. Initialize memory pointer as a source. 3. Initialize memory pointer as a destination. 4. Initialize counter. 5. Move the content of location in to accumulator. 6. Exchange with destination memory 7. Move the content of accumulator to first source location.	**Algorithm:**

8. Increment source location. 9. Increment destination location. 10. Decrement iteration count and if not zero jump to step 5. 10. Stop	

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV R0,#50H	Source memory pointer		
MOV R1,#60H	Destination memory pointer		
MOV R7,#05H	Set counter		
UP:MOV A,@R0	Get the number from source into accumulator		
XCH A,@R1	Exchange with destination		
MOV @R0,A	Move it into destination		
INC R0	Point on next source memory location		
INC R1	Point on next destination memory location		
DJNZ R7,UP	Repeat the transfer till counter R7 =0		
END	End of Program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Before execution				After execution			
Memory Address	**Contents**	**Memory Address**	**Contents**	**Memory Address**	**Contents**	**Memory Address**	**Contents**

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write assembly program to exchange 5 data bytes from internal memory locations using MOV instructions.
2. Write difference between XCH and XCHD instruction.
3. Explain EQU and DB directive with example.
4. Write difference between MOVX and MOVC instruction.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.20: Assembly language program to find smallest number from the given data bytes stored in internal / external data memory locations.

I Practical Significance

Finding smallest and largest number in a given array is common operation used in many algorithms and applications such as data searching and sorting. This practical will help the students to develop skills to compare numbers in an array.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program to find smallest number from the given data.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

CJNE destination, source, relative address

In 8051, the actions of comparing and jumping are combined into a single instruction called CJNE. It compares two operands and jump to relative address if they are not equal. In addition, it changes the CY flag to indicate if the destination operand is larger or smaller. This instruction is used to find the largest/smallest number.

DJNZ Rn, relative address

These instruction decrements the contents of register and jump to the relative address if not zero.

Both of these instructions are used in to find the largest/smallest number program.

To find smallest number is the process of comparing content of group of memory location, after finding smallest number result is placed in the memory.

MEMORY WINDOW:
Before execution:
D:0x40H: 22 AB 3D 44 55

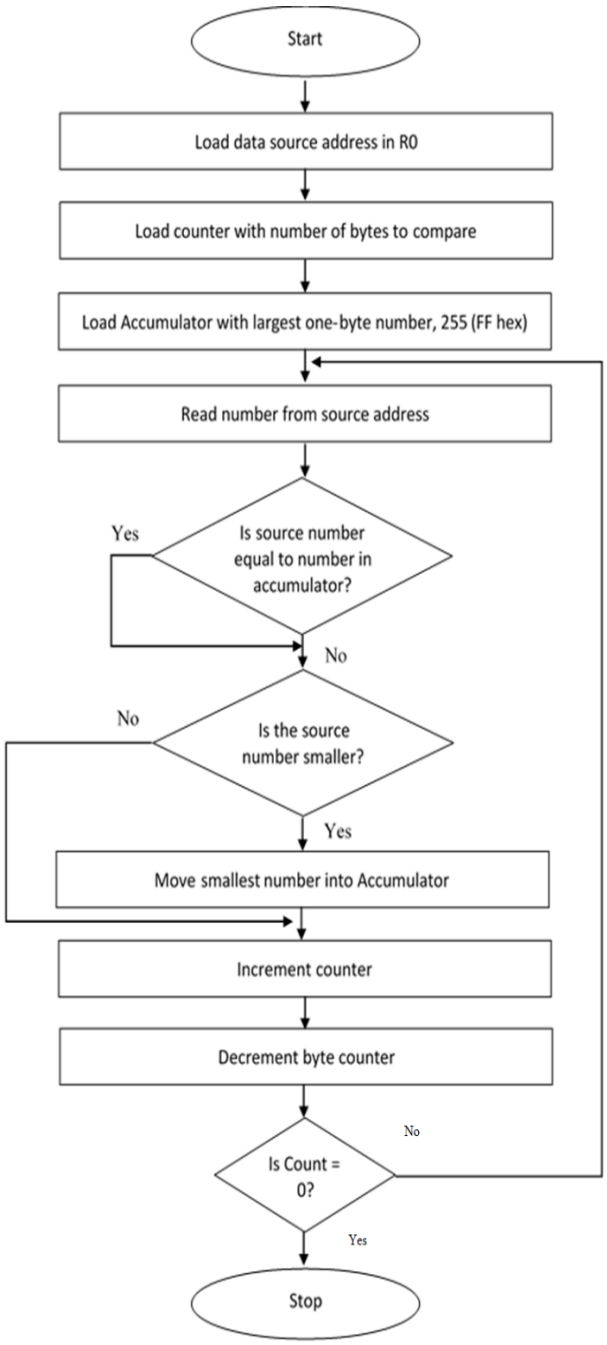
D:0X50H: 00 00 00 00 00

After execution:
D:0x40H: 22 AB 3D 44 55

D:0X50H: 22 00 00 00 00

Sample program: Write ALP to find smallest number from the given FIVE bytes stored in internal memory locations 40H onwards and store the result in location 50H.	**Student activity: Write ALP to find smallest number from the given TEN bytes stored in internal memory locations 50H onwards and store the result in location 60H.**
Algorithm: 1. Initialize source pointer R0 to 40H. 2. Initialize byte counter. 3. Load Accumulator with largest one byte number. 4. Move the contents of source location to B register. 5. Compare the two numbers. 6. If number is less then next number then go to step 8. 7. Replace number with next number which is largest. 8. Increment memory pointer to read next number in the array. 9. Decrement byte counter by 1. 10. If byte counter is not zero go to step 4. 11. Store the result. 12. Stop	**Algorithm:**

Flowchart:



Flowchart

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV R0, #40H	Initialize source pointer R0 to 40H		
MOV R1, #05H	Initialize byte counter		
MOV A, #0FFH	Load maximum value for comparison		
UP: MOV B, @R0	Read first byte to B register		
CJNE A, B, DOWN	Compare first byte to max value if not equal go to down		
DOWN: JC A_SMALL	Check carry		
MOV A, B	Move small value to A		
A_SMALL: INC R0	Increment the contents of R0		
DJNZ R1, UP	Decrement counter by one Is it zero? No, jump to UP		
MOV 50H, A	Store smallest number to 50H		
END	End of Program		

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 0r 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output in the register/memory.
11. Note down the readings in observation table

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

…….

..……

……..

XIII Observation Table

Before execution		After execution	
Memory location	Data	Memory location	Data
40H		50H	
41H			
42H			
43H			
44H			

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions to ensure the achievement of identifies CO.

1. Write assembly language program to count zeros in a data 86H.
2. Write assembly language program to find the average of five numbers.
3. Write any three-manufacturing company names those are producing 8051microcontrollers.
4. Explain the significance of program status word (PSW).

[Space for Answers]

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.21: Assembly language program to find largest number from the given data bytes stored in internal / external data memory locations.

I Practical Significance

Finding smallest and largest number in a given array is common operation used in many algorithms and applications, such as finding the maximum value in a data set or arranging the numbers in ascending and descending order. This practical will help the students to develop skills to use the compare and loop instructions.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program to find largest number from the given data.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

CJNE destination, source, relative address

In 8051, the actions of comparing and jumping are combined into a single instruction called CJNE. It compares two operands and jump to relative address if they are not equal. In addition, it changes the CY flag to indicate if the destination operand is larger or smaller. This instruction is used to find the largest/smallest number.

DJNZ Rn, relative address

These instruction decrements the contents of register and jump to the relative address if not zero.
Both of these instructions are used to find the largest/smallest number.

To find largest number is the process of comparing content of group of memory location, after finding largest number result is placed in the memory.

MEMORY WINDOW:
Before execution:
D:0x40h: 22 AB 3D 44 55

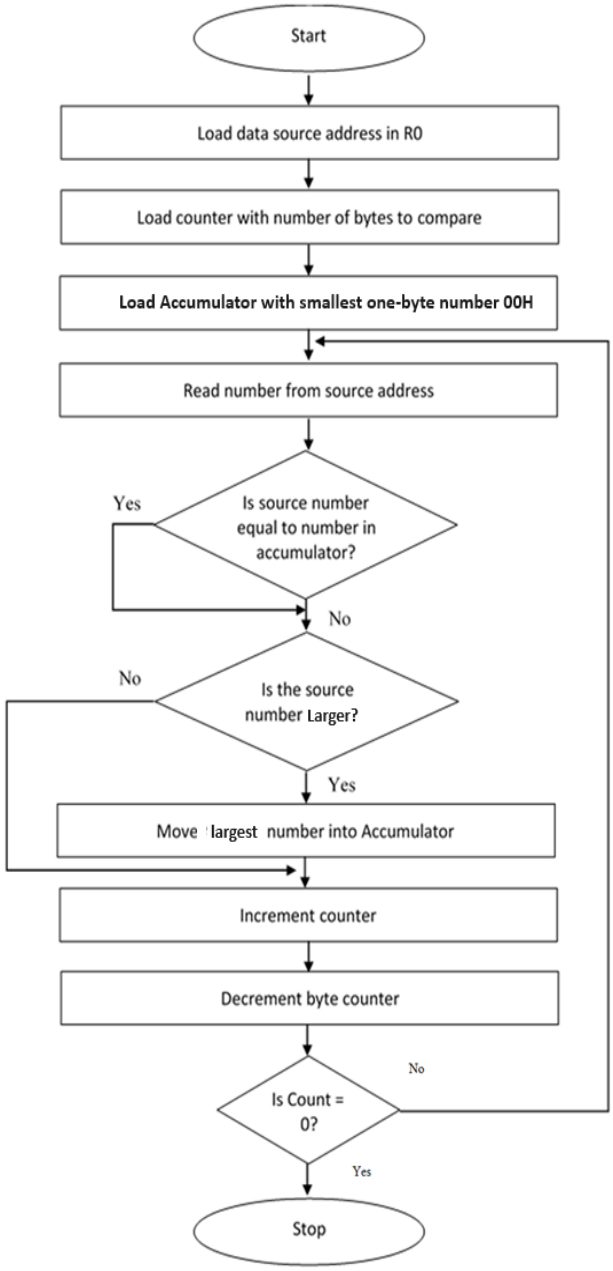
D:0x50h: 00 00 00 00 00

After execution:
D:0x40h: 22 AB 3D 44 55

D:0x50h: AB 00 00 00 00

Sample program: Write ALP to find largest number from the given FIVE bytes stored in internal memory locations 40H onwards and store the result in location 50H.	**Student activity: Write ALP to find largest number from the given TEN bytes stored in internal memory locations 50H onwards and store the result in location 60H.**
Algorithm: 1. Initialize source pointer R0 to 40H. 2. Initialize byte counter of 5 bytes. 3. Load Accumulator with smallest one byte number. 4. Move the contents of source location to B register. 5. Compare the two numbers. 6. If number is greater then next number then go to step 8. 7. Replace number with next number which is smallest. 8. Increment memory pointer to read next number in the array. 9. Decrement byte counter by 1. 10. If byte counter is not zero go to step 4. 11. Store the result. 12. Stop	**Algorithm:**

Flowchart:	**Flowchart**



Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV R0, #40H	Initialize source pointer R0 to 40H		
MOV R1, #05H	Initialize byte counter		
MOV A, #00H	Clear A register		

UP: MOV B, @R0	Read first byte to B register		
CJNE A, B, DOWN	Compare first byte to lowest value		
DOWN: JNC A_LARGE	Check carry		
MOV A, B	Move large value into A register		
A_LARGE: INC R0	Increment the contents of R0 pointer		
DJNZ R1, UP	Decrement counter by one Is it zero? No, jump to UP		
MOV 50H, A	Store largest number to 50H		
END	End of Program		

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 0r 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.

10. Observe the output in the register/memory.
11. Note down the readings in observation table

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Before execution		After execution	
Memory location	Data	Memory location	Data
40H		50H	
41H			
42H			
43H			
44H			

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. State importance of carry flag in the PSW register to find smallest or largest number.
2. Write a program to
   a) Load the accumulator with the value 55H, and b) complement the A register 50 times.
3. Write two conditional and unconditional jump instructions.
4. Write appropriate instruction to increment the content of 30H memory location.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.22: Assembly language program for arranging numbers in ascending order stored in external memory locations.

I Practical Significance

Sorting is any process of arranging information systematically in ascending or descending order. This allows us to write better programs like indexing to fetch the information faster, allows faster search techniques, removes duplicate information and has many uses in statistical applications. This practical will help the students to develop skills to understand how to access data from external memory and use of branch instructions.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program for arranging numbers in ascending order

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Ascending order/Descending order

The block of data consists of numbers in random order, to arrange these numbers in ascending or descending order bubble sort method is used. If the given block of data has to be sorted in ascending order, then bubble sort will start by comparing the first element of the block with the second element, if the first element is greater than the second element, it will swap both the elements, and then move on to compare the second and the third element, and so on.

Byte counter –to access data from block of data

Pass counter –to repeat this comparison are required to arrange the numbers in ascending or descending order.

Branch instructions like JNC—jump if no carry and CJNE—compare and jump if not equal to are used to sort the numbers.

External data memory is read/write. Since external data memory is indirectly accessed through a data pointer register (which must be loaded with an address), it is slower than access to internal data memory.

DPTR is used to point to data. It is used by a number of commands which allow the 8051 to access external memory. When the 8051 accesses external memory it will access external memory at the address indicated by DPTR.

MOVX instruction

MOVX is a widely used instruction allowing access to external data memory space. To bring externally stored data into the CPU, we use the instruction “MOVX A, @DPTR”. This instruction will read the byte of data pointed to by register DPTR and store it in the accumulator.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 0r 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output in the register/memory.
11. Note down the values of external memory location in the observation table.

Sample program: Write ALP for arranging FIVE numbers in ascending order stored in external memory location 3000H.	**Student activity: Write ALP for arranging TEN numbers in ascending order stored in external memory location 4000H.**
Algorithm: 1. Initialize comparison or pass counter 2. Initialize memory pointer DPTR to read number from array 3. Initialize byte counter 4. Read numbers from the array 5. Compare two numbers. 6. If number <=next number, then go to step 9. 7. Interchange or swap numbers. 8. Increment memory pointer to read next number from array. 9. Decrement byte counter by one. 10. If word counter is not equal to zero, then go to step 2. 11. Stop	**Algorithm:**

Assembly language program

Instructions	**Comments**	**Instructions**	**Comments**
ORG 0000H	Start of program		
UP1: MOV R0, #05H	Initialize pass counter		
MOV DPTR, #3000H	Initialize memory pointer		
MOV R1, #04H	Initialize byte counter		
UP: MOV R2, DPL	Save the lower byte address		
MOVX A, @DPTR	Read number from array		
MOV B, A	Transfer the number to B register		
INC DPTR	Increment memory pointer		
MOVX A, @DPTR	Read next number from array		
CJNE A, B, DN	Compare number with next number		
AJMP SKIP	Jump to label SKIP		
DN: JNC SKIP	If number<next number then go to SKIP		
MOV DPL, R2	Else exchange the number with next number		

Instructions	Comments	Instructions	Comments
MOVX @DPTR, A	Move A register data to external memory location pointed by DPTR		
INC DPTR	Increment data pointer		
MOV A, B	Move data from B register to A register		
MOVX @DPTR, A	Move A register data to external memory location pointed by DPTR		
SKIP: DJNZ R1, UP	Decrement byte and if count byte is not zero go to Up		
DJNZ R0, UP1	Decrement pass counter and if not zero go to UP1		
END	End of Program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Before execution		After execution	
Memory location	Data	Memory location	Data
3000H		3000H	
3001H		3001H	
3002H		3002H	
3003H		3003H	
3004H		3004H	
3005H		3005H	

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write instructions to increment the content of DPTR by one.
2. Explain the role of pins PSEN' ALE and EA' in accessing external memory.
3. Calculate the number of address lines required to access 8 KB RAM.
4. Intel 8051 Follows Which Architecture?

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://electronicsforyou.in/8051-program-to-arrange-numbers-in-ascending-order/
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.23: Assembly language program for arranging numbers in descending order stored in external memory locations.

I Practical Significance

Sorting is any process of arranging information systematically in ascending or descending order. This allows us to write better programs like indexing to fetch the information faster, allows faster search techniques, removes duplicate information and has many uses in statistical applications. This practical will help the students to develop skills to understand accessing data techniques from external memory.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program for arranging numbers in descending order

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Ascending order/Descending order

The block of data consists of numbers in random order, to arrange these numbers in ascending or descending order bubble sort method is used.

If the given block of data has to be sorted in ascending order, then bubble sort will start by comparing the first element of the block with the second element, if the first element is greater than the second element, it will swap both the elements, and then move on to compare the second and the third element, and so on.

Byte counter –to access data from block of data

Pass counter –to repeat this comparison are required to arrange the numbers in ascending or descending order.

Branch instructions like JC—jump if carry and CJNE—compare and jump if not equal to are used to sort the numbers.

External data memory is read/write. Since external data memory is indirectly accessed through a data pointer register (which must be loaded with an address), it is slower than access to internal data memory.

DPTR, as the name suggests, is used to point to data. It is used by a number of commands which allow the 8051 to access external memory. When the 8051 accesses external memory it will access external memory at the address indicated by DPTR.

MOVX instruction

MOVX is a widely used instruction allowing access to external data memory space. To bring externally stored data into the CPU, we use the instruction “MOVX A, @DPTR”. This instruction will read the byte of data pointed to by register DPTR and store it in the accumulator.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 0r 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output in the register/memory.
11. Note down the values of external memory location 3000H in the observation table.

Sample program: Write ALP for arranging FIVE numbers in descending order stored in external memory location 3000H.	**Student activity: Write ALP for arranging TEN numbers in descending order stored in external memory location 4000H.**
Algorithm: 1. Initialize comparison or pass counter 2. Initialize memory pointer to read number from array 3. Initialize byte counter 4. Read numbers from the array 5. Compare two numbers. 6. If number >=next number, then go to step 9. 7. Interchange or swap numbers. 8. Increment memory pointer to read next number from array. 9. Decrement byte counter by one. 10. If word counter is not equal to zero, then go to step 2. 11. Stop	**Algorithm:**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
UP1: MOV R0, #05H	Initialize pass counter		
MOV DPTR, #3000H	Initialize memory pointer		
MOV R1, #04H	Initialize byte counter		
UP:MOV R2, DPL	Save the lower byte address		
MOVX A, @DPTR	Read number from array		
MOV B, A	Transfer the number to B register		
INC DPTR	Increment memory pointer		
MOVX A, @DPTR	Read next number from array		
CJNE A, B, DN	Compare number with next number		
AJMP SKIP	Jump to label SKIP		
DN: JC SKIP	If number>next number then go to SKIP		
MOV DPL, R2	Else exchange the number with next number		
MOVX @DPTR, A	Move A register data to external memory location pointed by DPTR		
INC DPTR	Increment data pointer		
MOV A, B	Move data from B register to A.		
MOVX @DPTR, A	Move A register data to external memory location pointed by DPTR		
SKIP: DJNZ R1, UP	Decrement byte and if count byte is not zero go to Up		
DJNZ R0, UP1	Decrement pass counter and if not zero go to UP1		
END	End of Program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……..

..……

……..

XIII Observation Table

Before execution		After execution	
Memory location	Data	Memory location	Data
3000H		3000H	
3001H		3001H	
3002H		3002H	
3003H		3003H	
3004H		3004H	
3005H		3005H	

XIV Result(s)

……..

……..

..……

……..

XV Interpretation of results

……..

.....……..

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Write a program to (a) load the accumulator with the value 55H, and (b) complement the ACC 70 times.
2. Explain JC code address and JNC code address instruction with example
3. Write any four Keil IDE features.
4. State the status of carry flag after comparison operation,
   a) A<B register b) A>B register

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.24: Assembly language program MASK and SET particular bit of given register using 1) bit addressable instructions 2) Logical instructions.

I Practical Significance

Source code that does bit manipulation makes use of the bitwise operations: AND, OR, XOR, NOT, and possibly other operations analogous to the Boolean operators; there are also bit shifts and operations to count ones and zeros, find high and low one or zero, set, reset and test bits, extract and insert fields, mask and zero fields, gather and scatter bits to and from specified bit positions or fields. Most microcontrollers frequently deal with bits of data rather than bytes. This practical will help the students to develop skills to understand how to access single bit instructions.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcomes

1. Develop and execute an assembly language program for masking particular bit of given register.
2. Develop and execute an assembly language program to SET particular bit of given register.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

The 8051 microcontrollers can perform single bit operations. The operations include set, clear, AND, OR and complement instructions. Also included bit–level moves or conditional jump instructions. All bit accesses instructions use direct addressing mode.

Logical instructions perform standard Boolean operations on byte operations such as AND, OR, XOR, NOT (compliment). Other logical operations are clear accumulator, rotate accumulator left and right, and swap nibbles in accumulator.

Program explanation

Initial Value: MOV A, #0FFH loads the accumulator with 0xFF.

Mask to Clear a Bit: ANL A, #0FBH uses the AND logical instruction to clear the 3rd bit.

The mask 0xFB (1111 1011) has all bits set to 1 except the 3rd bit.

Mask to Set a Bit: ORL A, #10H uses the OR logical instruction to set the 5th bit. The mask 0x10 (0001 0000) has the 5th bit set to 1.

VII Actual Circuit diagram used in laboratory with related equipment rating.

Note: This practical is based on Integrated Design Environment (IDE) software, not required circuit diagram.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 0r 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

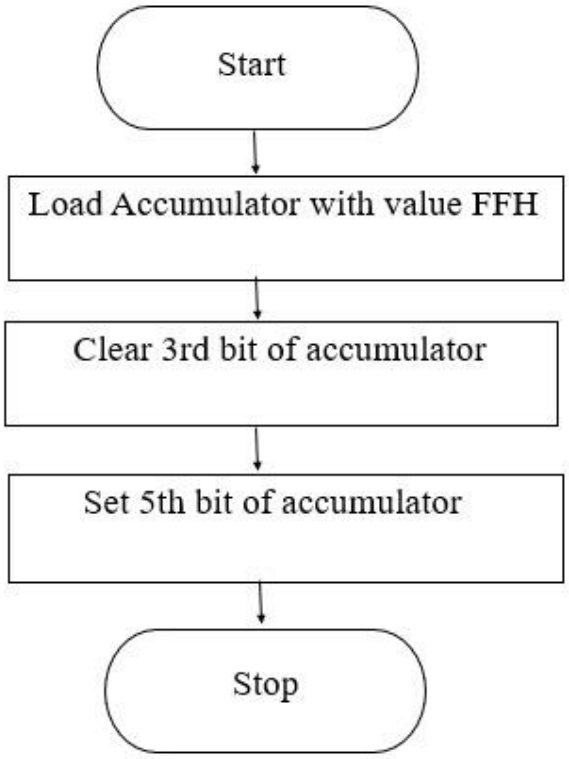
6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output in the register/memory/ Input output port.
11. Note the contents of the registers/memory/Input output port in observation table.

Sample program: Write assembly program to Mask (clear) the 3rd bit of accumulator and set the 5th bit of accumulator	**Student activity: Write assembly program to Mask (clear) the 3rd bit of Port 1 and set the 5th bit of Port 1**
Algorithm: 1. Start of program 2. Load accumulator with a sample value, say 0xFF 3. Clear 3rd bit of accumulator (make it 0) FBH = 1111 1011 4. Set 5th bit of accumulator (make it 1) 10H = 0001 0000 5. End of program	**Algorithm:**

Flowchart



Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV A, #0FFH	Load accumulator with a sample value, say 0xFF		
ANL A, #0FBH	Clear 3rd bit of accumulator (make it 0) FBH = 1111 1011		
ANL A, #10H	Set 5th bit of accumulator (make it 1) 10H = 0001 0000		
END	End of program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……..

..……

……..

XIII Observation Table

Sr. No.	Operation	Result
1	Mask (clear) the 3rd bit of accumulator	
2	Set the 5th bit of accumulator	

XIV Result(s)

……..

……..

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identifies CO.

1. List any two Input output port related instructions of 8051 microcontroller.
2. To mask certain bits of the accumulator we must ANL it with_____.

3. Find the contents of register A after execution of the following code.
   CLR A
   ORL A, #99H
   CPL A
4. Indicate which of the following registers are bit addressable.
   a) A b) B c) R4 d) PSW

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.25: Assembly language program to get a rolling display on port 2.

I Practical Significance

The 8051 microcontrollers have 4 input and output ports each of 8-bit. Ports can be configured as input or output. Ports exchange the data and signals with external devices. This practical will help the students to develop skills to handle I/O function for exchanging data and signals between external devices and a microcontroller.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcomes

1. Access various registers in 8051 microcontroller.
2. Develop and execute programs in assembly language for microcontroller.

IV Laboratory Learning Outcome

Develop and execute an assembly language program to get a rolling display on given I/O port.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

The program begins by setting the content of Port 2 so that the first LED is on. This is done by moving 01H to Port 2 (MOV P2, #01H).

Main Loop: The main loop of the program consists of a delay subroutine call to pace the shifting speed, followed by a rotate left instruction (RL A) that shifts the bits of the accumulator.

Rolling Effect: The RL A instruction effectively rotates all bits to the left, and the leftmost bit wraps around to the rightmost position, creating a rolling effect.

Delay Subroutine: This subroutine introduces a delay between each shift to make the rolling visible. The delay is achieved through nested loops (DELAY1 and DELAY2), which decrement register values until they reach zero.

Continuous Looping: The SJMP MAIN_LOOP instruction ensures that the program continues indefinitely.

This program should provide a simple rolling display on Port 2 of the 8051 microcontrollers, where the LEDs connected to the port will light up one after the other in sequence.

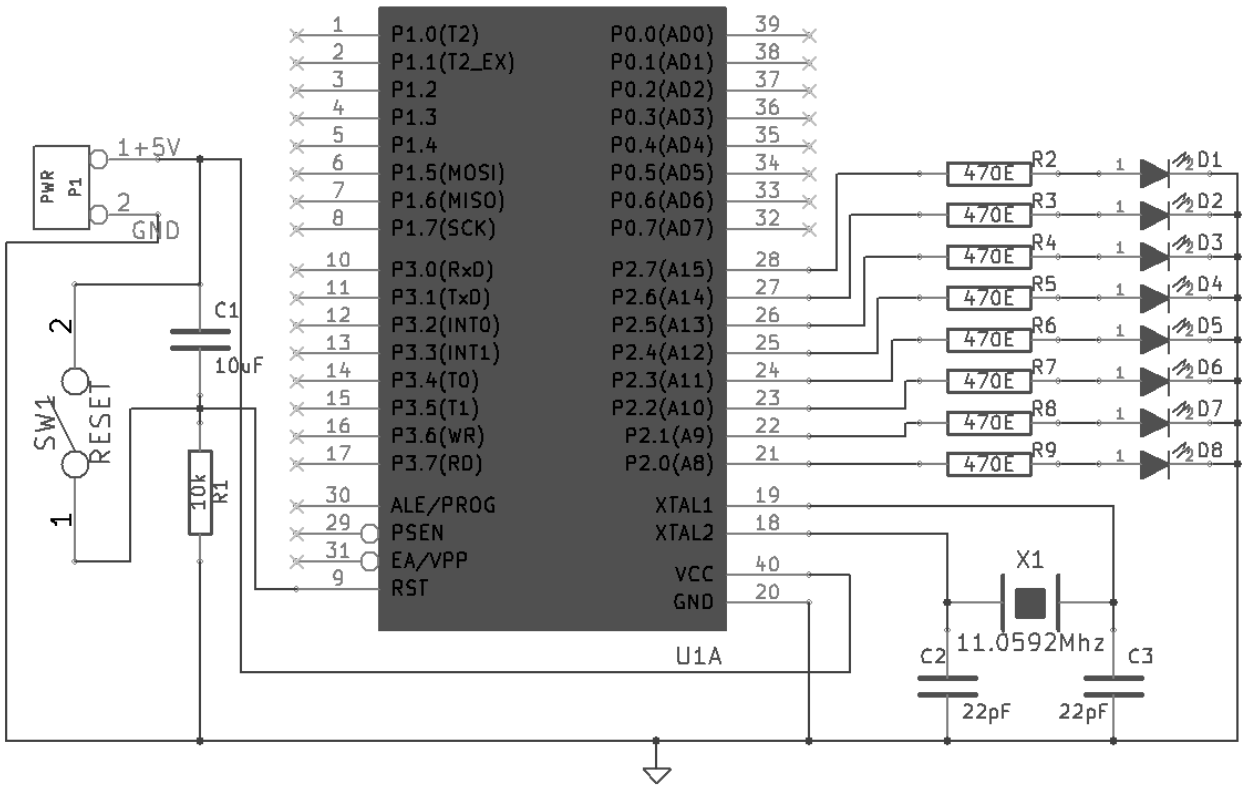


Fig. 25.1 8051 connection with 8 LED's

Courtesy: https://exploreembedded.com/wiki/7.8051_Interfacing:LEDs

VII Actual Circuit diagram used in laboratory with related equipment rating.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

Write Program

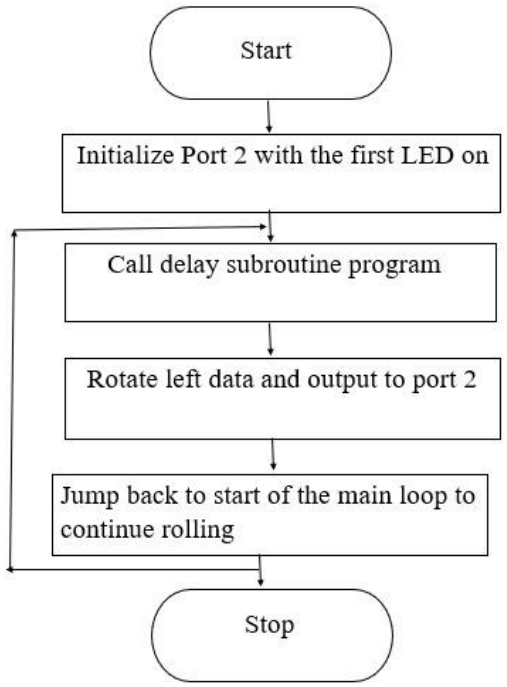
1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select AT89C51 0r 80c51AH.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output in the register/memory/Input output port.
11. Note the contents of the registers/memory/Input output port in observation table.

Sample program: Write assembly program to get a rolling display on port 2.	**Student activity: Write assembly program to get a rolling display on port 1 and port 2 simultaneously.**
Algorithm: 1. Start of program 2. Initialize Port 2 with the first LED-ON 3. Call delay subroutine for visible shifting effect 4. Rotate data left 5. Output to Port 2 6. Jump back to start of the main loop to continue rolling 7. Call delay subroutine 8. Outer loop counter 9. Inner loop counter 10. Decrement R1 until it's zero 11. Decrement R2 until it's zero 12. Return from subroutine 13. End of program	**Algorithm:**
Flowchart  	

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV A, #01H	Load A register with 01H value		
MOV P2, A	Send value to Port 2 with the first LED on (0000 0001)		
MAIN_LOOP: ACALL DELAY	Call delay subroutine for visible shifting effect		
RL A	Rotate left data		
MOV P2, A	Output to Port 2		
SJMP MAIN_LOOP	Jump back to start of the main loop to continue rolling		
DELAY: MOV R2, #20	Outer loop counter		
L1: MOV R1, #255	Inner loop counter		
L2: DJNZ R1, L2	Decrement R1 until it's zero		
DJNZ R2, L1	Decrement R2 until it's zero		
RET	Return from subroutine		
END	End of program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

…….

..……

……..

XIII Observation Table

Sr. No.	Operation	Result
1	Port 2 status after 2 times execution of RL A instruction	
2	Port 2 status after 4 times execution of RL A instruction	

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. For each of the following instructions, state to which port the bit belongs.
   a) SETB 86H b) CLR 87H c) SETB 92H d) SETB 0A7H
2. List any two Logical instructions of 8051 microcontroller.
3. Illustrate with example bit and byte handling one instruction in the 8051 microcontrollers.

[Space for Answers]

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

..……..…

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Observations and recording	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.26: Interfacing of LED and switch with 8051 to turn ON / OFF the LED.

I Practical Significance

Switch and LED are the basic examples of input and output device used in Industrial or domestic appliances. This practical will help the students to develop skills to interface LED and switch to microcontroller.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Use microcontroller in various applications.

IV Laboratory Learning Outcome(s)

1. Interface LED and switch with 8051.
2. Interface SWITCH with 8051.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Light Emitting Diodes are the semiconductor light sources. Commonly used LEDs will have a cut-off voltage of 1.7V and current of 10mA. When an LED is applied with its required voltage and current it glows with full intensity.

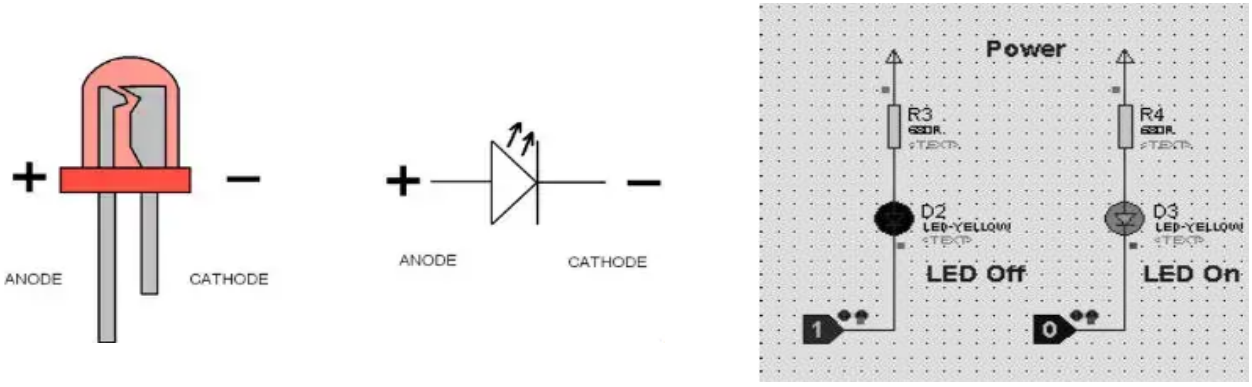


Fig. 26.1 LED symbol and connection logic

A push-button is a type of momentary switch which when pressed completes a circuit, and when released, the button returns to its original position and breaks or interrupts a circuit.

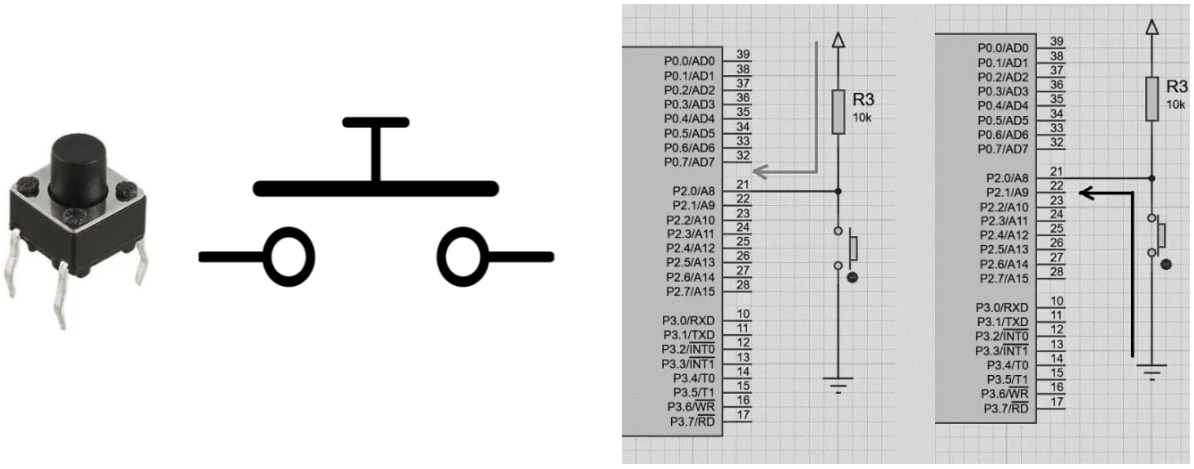


Fig. 26.2 Push to on switch symbol and connection logic

- When the switch is OFF (not pressed), the input to pin P2.0 is a HIGH pulse (1).
- When the switch is ON (pressed), the input to pin P2.0 is a LOW pulse (0).

Switch is interfaced using a negative logic with a 10k ohm pull-up resistor.

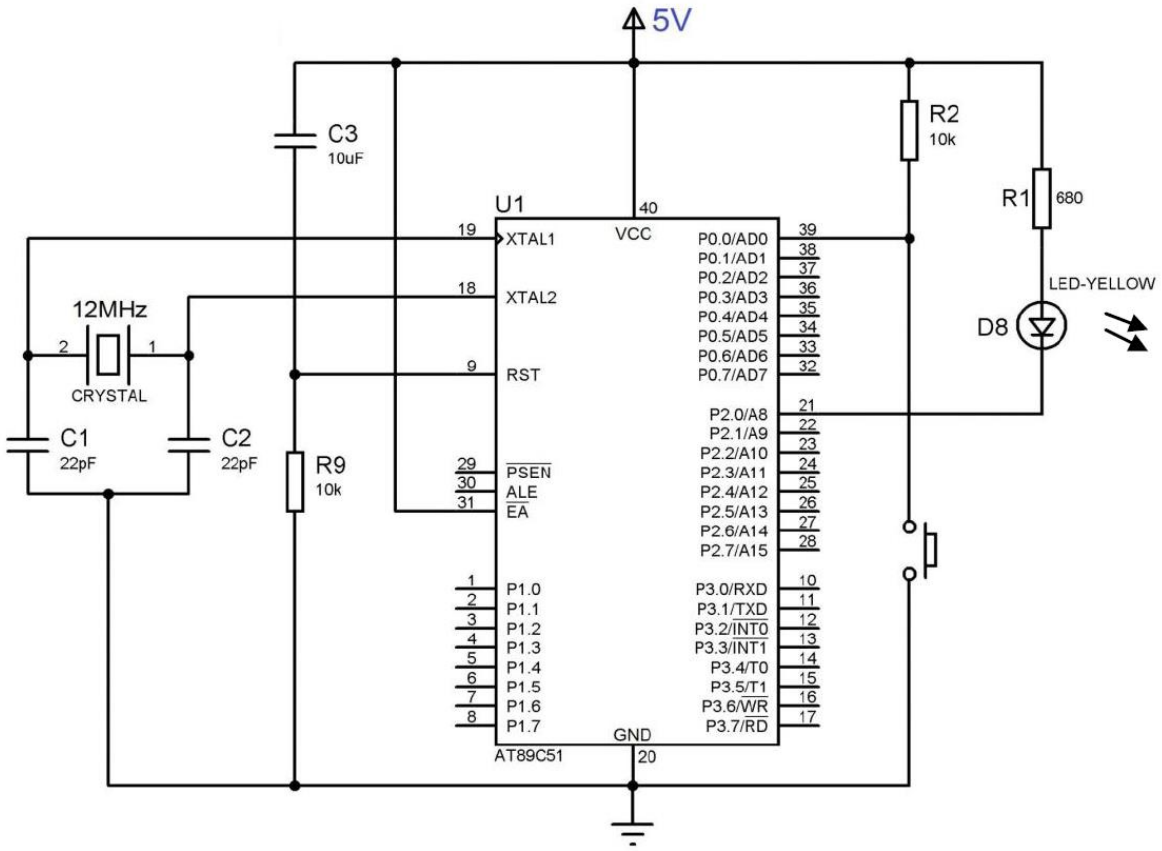


Fig. 26.3 8051 connection to switch and LED

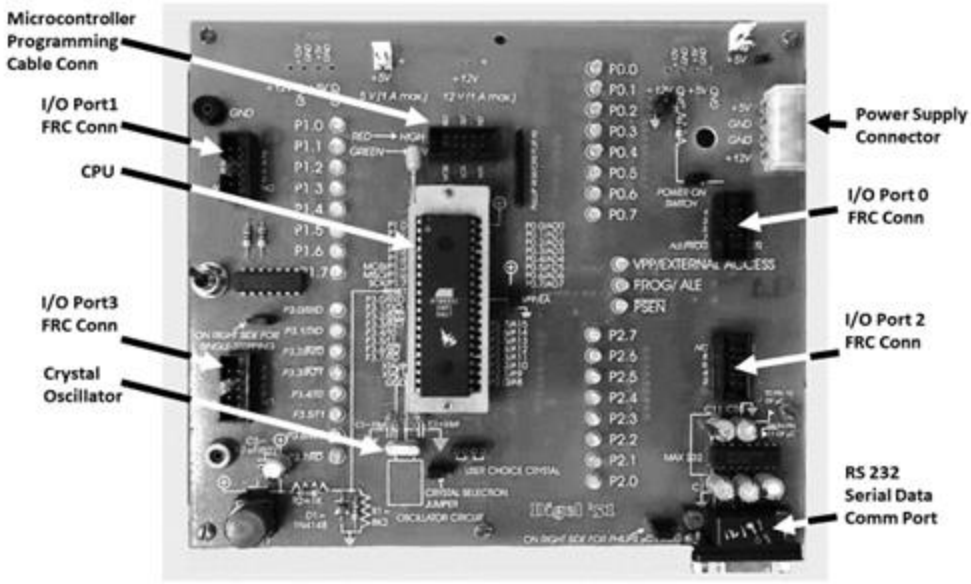


Fig. 26.4 8051 Development Board

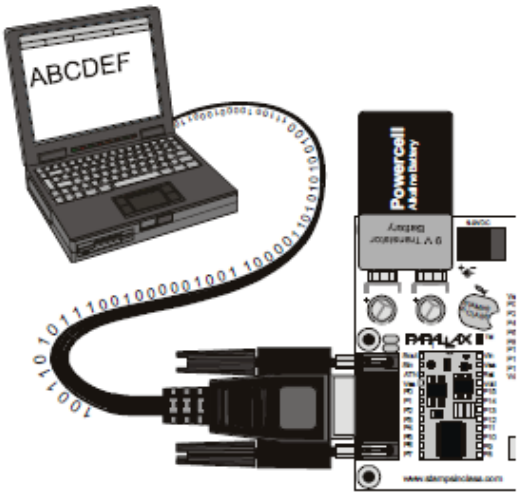
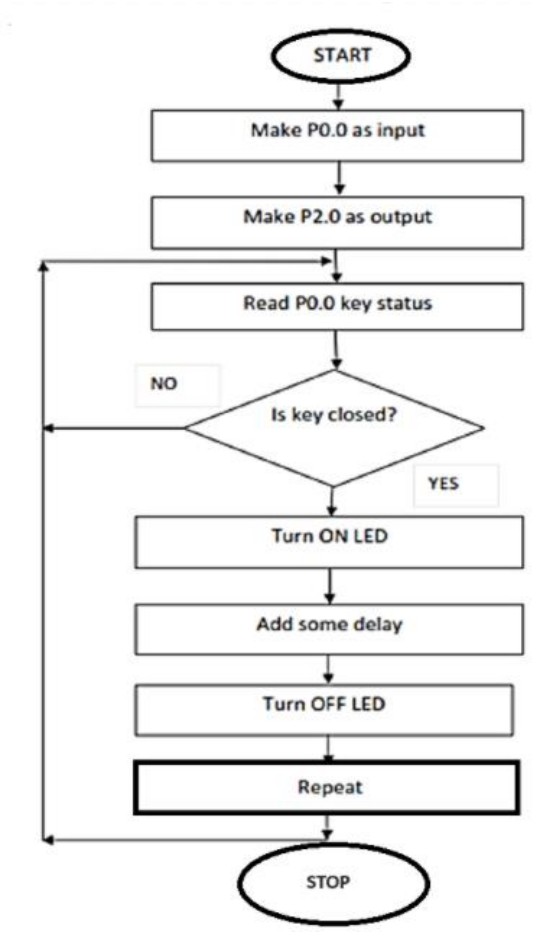


Fig. 26.5 8051 Programming through serial cable

Sample program: Write and execute assembly program to read the status of key and turn on LED connected to port pin of microcontroller and turn off after some delay.	**Student activity: Write and execute assembly program to turn on 8 LEDs connected to port 2 when switch is pressed.**
Algorithm: 1. Initialize port pin P0.0 as input pin 2. Make LED off initially 3. Monitor the status of switch 4. If switch is closed then turn ON the LED.	**Algorithm:**

5. Add some delay 6. Turn OFF the LED 7. Repeat from step 3	
Flowchart:  	**Flowchart**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
SETB P0.0	Define input pin		
CLR P2.0	Define output pin		
UP: SETB P2.0	Turn off LED		
HERE: JB P0.0, HERE	Monitor switch pin and wait here till switch is pressed		
CLR P2.0	Turn on LED		
ACALL DELAY	Call delay subprogram		
SJMP UP	Jump to label UP		
DELAY: MOV R2, #200	Delay subprogram		
L2: MOV R3, #100	Load R3 register with count value 100		
L1: DJNZ R3, L1	Decrement R3 counter till it		

	becomes zero		
DJNZ R2, L2	Decrement R2 counter till it becomes zero		
RET	Return to main program		
END	End of the program		

VII Actual Circuit diagram used in laboratory with related equipment rating.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1
3	Microcontroller kit	Single board systems with 8KRAM, ROM memory with battery backup, 16X2 LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler, RS-232, USB interfacing facility with built in power supply.	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

X Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Keil IDE or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface Switch and LED to microcontroller as per circuit diagram shown in Figure 26.3
9. Press switch and observe LED On/Off status.

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

…….

..……

……..

XIII Observation Table

Sr. No.	Operation	Result
1	Switch closed	LED status-
2	Switch open	LED status-

XIV Result(s)

……..

…….

..……

……..

XV Interpretation of results

……..

.....…….

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identifies CO.

1. Two switches are connected to pin P1.0 and P1.1 with two LED's to P2.0 and P2.1 respectively. Write assembly program to turn ON LED1 when switch1 is pressed and turn ON LED2 when switch2 is pressed.
2. Write assembly program to toggle LED connected to port pin P1.7 continuously
3. Write specifications of two different types of switches and LED's available in the market.
4. Upon RESET, all the bits of ports are configured as?

[Space for Answers]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVIII References/Suggestions for further reading

1. http://www.circuitstoday.com/getting-started-with-keil-uvision
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Identifying components on developer kit	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.27: Interfacing of RELAY with 8051 to turn ON / OFF the LED.

I Practical Significance

In Industrial applications low power devices microcontrollers drive relays are used to control electrical loads beyond their direct drive capability. Relays are used wherever it is necessary to control a high power or high voltage circuit with a low power circuit, especially when isolation is desirable. This practical will help the students to develop skills to interface relay to microcontroller and turn it ON and OFF.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Use microcontroller in various applications.

IV Laboratory Learning Outcome

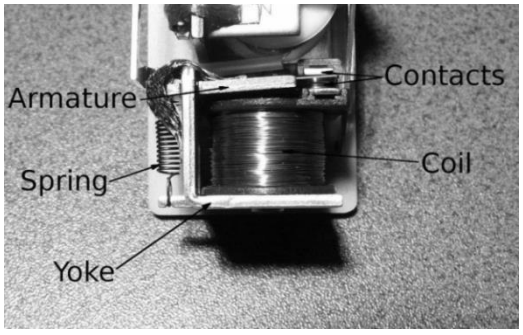
Interface RELAY with 8051.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

When an electric current is passed through the coil it generates a magnetic field that activates the armature and the consequent movement of the movable contact either makes or breaks (depending upon construction) a connection with a fixed contact.



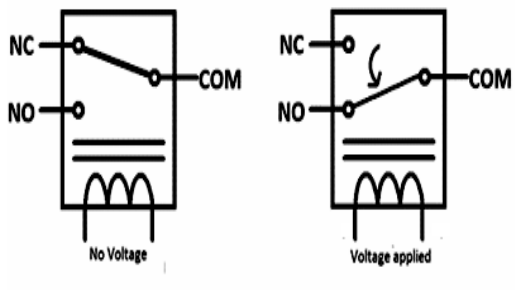


Fig 27.1 Simple electromechanical relay and SPDT Relay

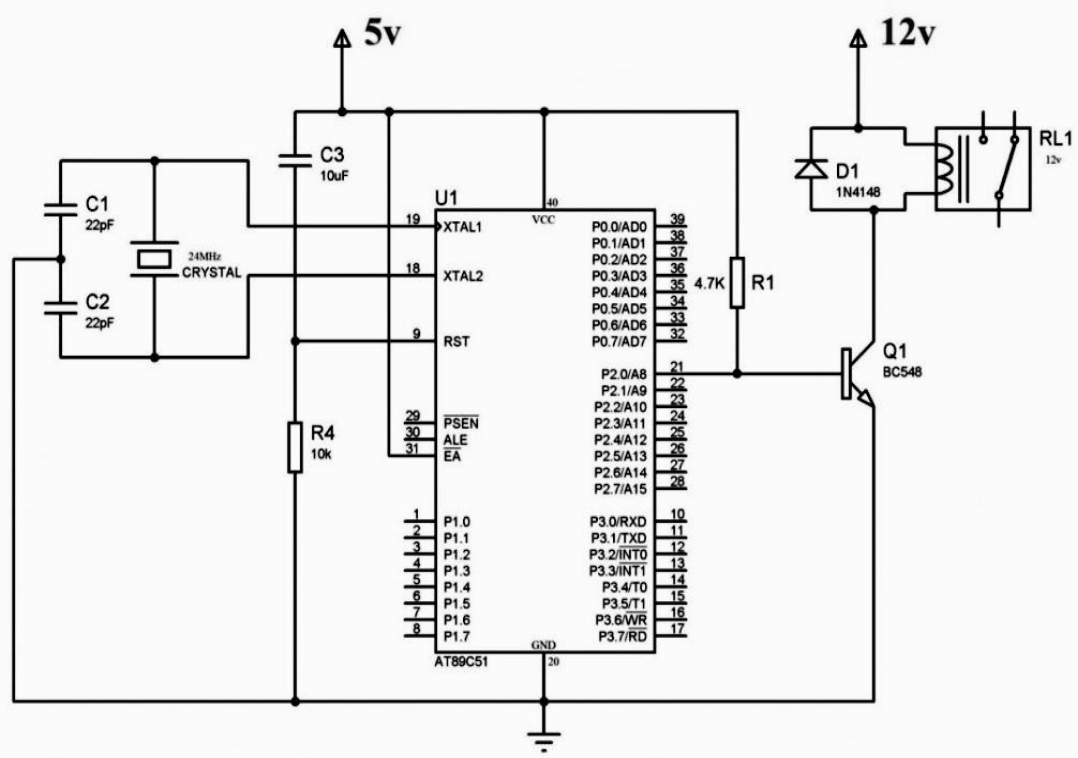


Fig 27.2 8051 connection to Relay

VII Actual Circuit diagram used in laboratory with related equipment rating.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1
3	Microcontroller kit	Single board systems with 8KRAM, ROM memory with battery backup, 16X2 LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler, RS-232, USB interfacing facility with built in power supply.	1
4	Relay	Realy driver circuit	1

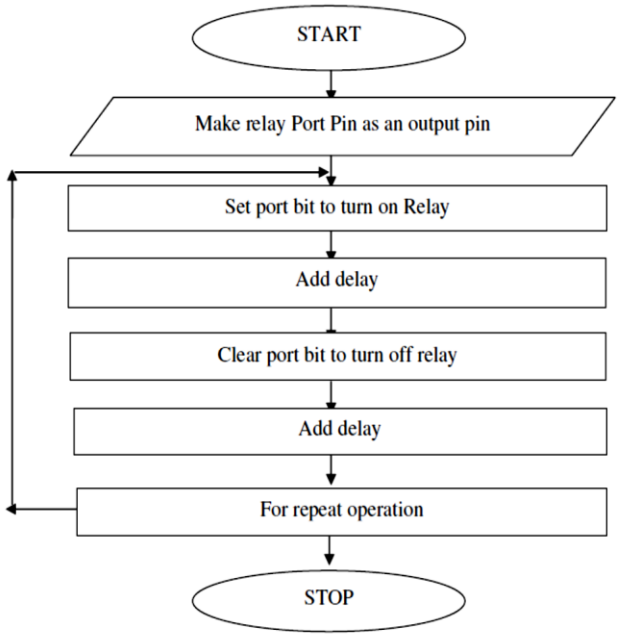
IX Precautions to be followed

1. Use always driver circuit before interfacing relay to the microcontroller.
2. Use fly back diode to avoid voltage spikes.

X Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface relay to the microcontroller as per circuit diagram shown in Fig 27.2
9. Observe the relay operation by connecting electrical load and note it in observation Table.

Sample program: Write assembly program to turn ON/OFF relay.	**Student activity:**
Algorithm: 1. Make the Port pin P2.0 used to Interface relay as an output pin. 2. Turn on relay by setting port bit. 3. Add delay 4. Turn off relay by clearing bit	**Algorithm:**

5. Add delay 6. For repeat operation go to step 2 **7.** Stop	
Flowchart:  	**Flowchart**

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MAIN: SETB P2.0	Turn ON Relay		
ACALL DELAY	Call delay sub program		
CLR P2.0	Turn OFF Relay		
ACALL DELAY	Call delay sub program		
SJMP MAIN	Jump to MAIN		
DELAY:MOV R3, #100	Delay sub program		
L3: MOV R4, #255	Set counter of 255		
L2: MOV R5, #255	Set counter of 255		
L1: DJNZ R5, L1	Decrement R5 counter till it becomes zero		
DJNZ R4, L2	Decrement R4 counter till it becomes zero		
DJNZ R3, L3	Decrement R3 counter till it becomes zero		
RET	Return to main program		
END	End of the program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Sr. No.	Port Pin Status Logic 1 (+5V) / Logic 0 (0V)	Relay Status ON/OFF
1		
2		

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions to ensure the achievement of identifies CO.

1. Explain what will happen if driver circuit is not connected between microcontrollers and relay.
2. Draw the Interfacing diagram of microcontroller with relay using ULN 2003A IC.
3. Give the advantages of a solid-state relay over an electromagnetic relay.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://en.wikipedia.org/wiki/Relay
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Identifying components on developer kit	20%
4	Follow ethical practices	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.28: Interfacing of 7-segment display with 8051 to give output as decimal number from 0 to 9.

I Practical Significance

In electronic displays like pricing menu at petrol pump, in metros, digital clocks and in many electronics appliances most commonly used display device is a 7-segment display. This practical will help the students to develop skills to interface 7-segment display to microcontroller and display number from 0 to 9.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Use microcontroller in various applications.

IV Laboratory Learning Outcome

Interface 7-segment display with 8051.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

7-segment display may use a light-emitting diode (LED) for each segment, or other light-generating or controlling techniques such as cold cathode gas discharge, vacuum fluorescent, incandescent filaments, and other. Types of LED based 7-segment display are i) Common Cathode Display ii) Common Anode. Display

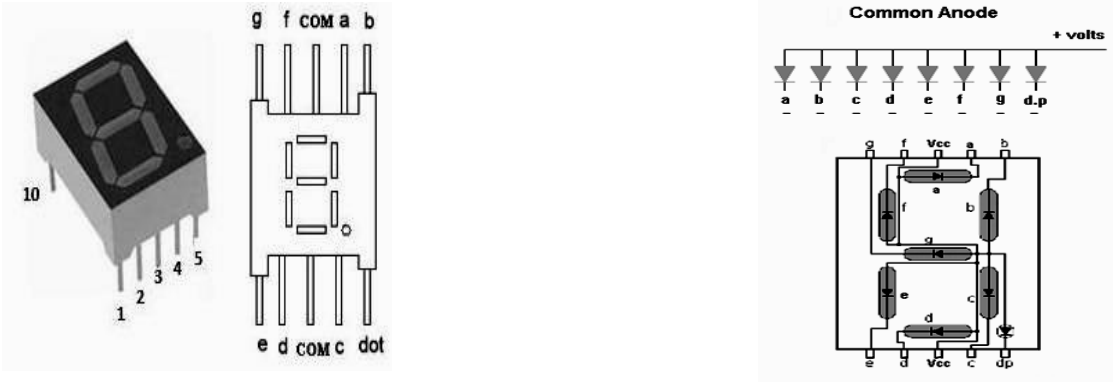


Fig 28.1 Common Anode Seven segment display and pin configuration

LED segment is ON for logic ‘0’ (Ground) and OFF for logic ‘1’ (+5V).

Table 28.1 Common Anode display number values

	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
Digit	**DP**	**G**	**F**	**E**	**D**	**C**	**B**	**A**	**Hex code**
0	1	1	0	0	0	0	0	0	0C0 H
1	1	1	1	1	1	0	0	1	0F9 H
2	1	0	1	0	0	1	0	0	0A4 H
3	1	0	1	1	0	0	0	0	0B0 H
4	1	0	0	1	1	0	0	1	99 H
5	1	0	0	1	0	0	1	0	92 H
6	1	0	0	0	0	0	1	0	82 H
7	1	1	1	1	1	0	0	0	0F8 H
8	1	0	0	0	0	0	0	0	80 H
9	1	0	0	1	0	0	0	0	90 H

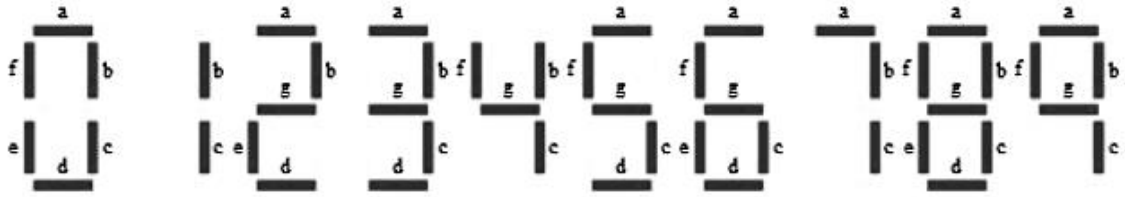


Fig 28.2 Seven segment display number patterns

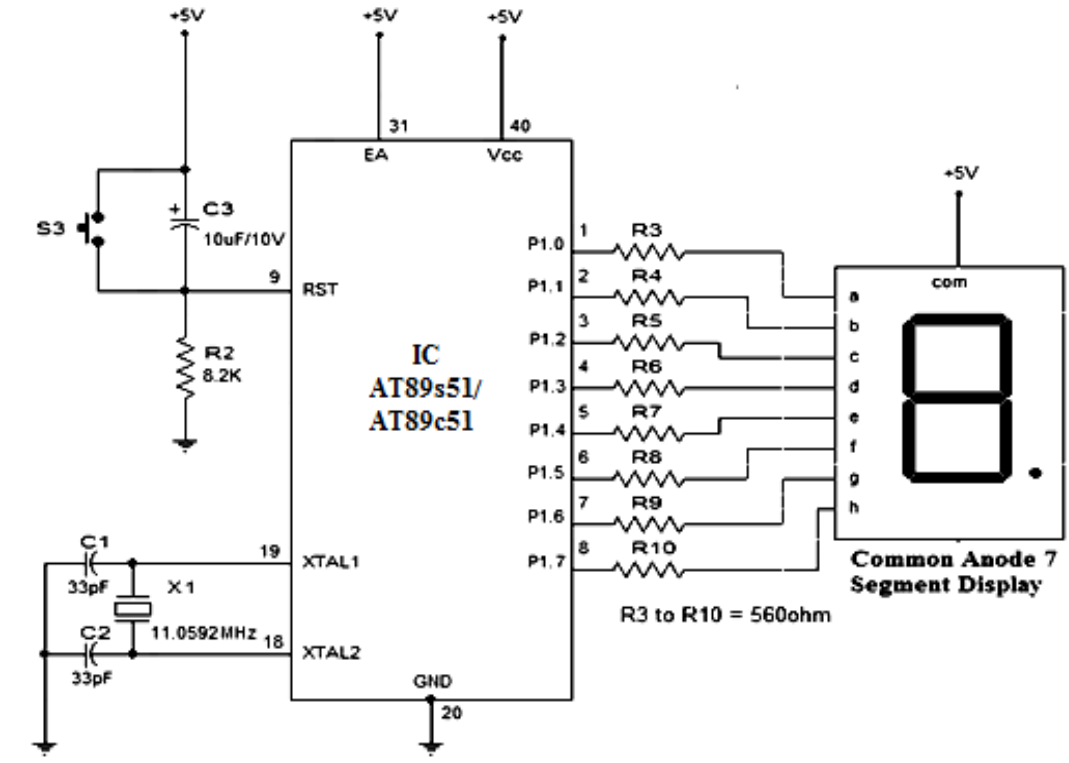


Fig 28.3 8051 connection to Common Anode seven segment display

VII Actual Circuit diagram used in laboratory with related equipment rating.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1
3	Microcontroller kit	Single board systems with 8KRAM, ROM memory with battery backup, 16X2 LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler, RS-232, USB interfacing facility with built in power supply.	1

IX Precautions to be followed

Check rules / syntax of assembly programming.

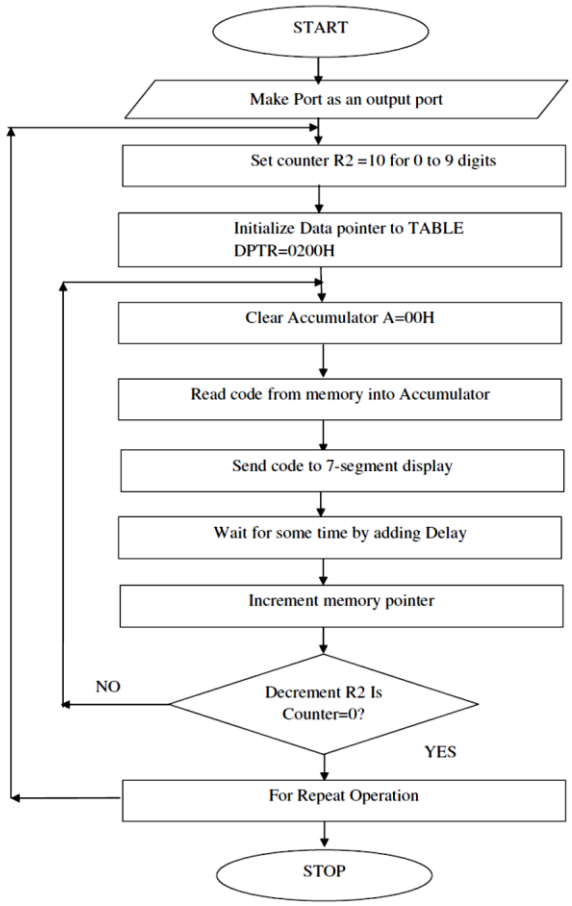
X Procedure

1. Write algorithm for given problem.
2. Draw flowchart.
3. Develop assembly program using Integrated Development Environment (Keil IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.

6. Create hex file.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface CA type 7 segment display to microcontroller as per circuit diagram shown in fig 28.3.
9. Observe and draw the display of numbers on 7-seven segment display.
10. Record the hex value in observation table.

Sample program: Write and execute a program to display decimal no 0 to 9 on seven-segment display.	**Student activity: Write and execute a program to display E and H on seven segment display.**
Algorithm: 1. Make the Port P1 used to Interface 7-segment display as an output port. 2. Set counter register R2 =10 for 0 to 9 digits. 3. Load DPTR with memory address where table is stored. 4. Clear Accumulator. 5. Read stored hex code of decimal digit from memory into Accumulator. 6. Send code to output port where 7-segment display is connected. 7. Increment memory pointer i.e. DPTR. 8. Decrement the counter register R2 and compare with 0 is counter =0? NO- go to step 4 to send next digit code. 9. For repeat operation go to step 2. 10. Stop	**Algorithm:**

Flowchart:



Flowchart

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV P1, #00H	Make port as output port		
REPEAT:MOV R2, #10	Set register as counter of 10 bytes		
MOV DPTR, #TABLE	Load address of memory into Data pointer		
UP: CLR A	Clear accumulator		
MOVC A, @A+DPTR	Read hex code from memory into accumulator		
MOV P1, A	Send hex code to port		
ACALL DELAY	Call DELAY subprogram		
INC DPTR	Increment memory pointer to read next digit hex code		
DJNZ R2, UP	Decrement counter and jump if not equal to zero to label UP.		
SJMP REPEAT	Repeat loop		
DELAY:MOV R3, #25	Delay Subroutine, Set		

	counter of 25		
L3:MOV R4, #100	Set counter of 100		
L2:MOV R5, #100	Set counter of 100		
L1: DJNZ R5, L1	Decrement R5 counter till it becomes zero		
DJNZ R4, L2	Decrement R4 counter till it becomes zero		
DJNZ R3, L3	Decrement R3 counter till it becomes zero		
RET	Return to main program		
ORG 0200H	Origin of TABLE		
TABLE:DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H	Decimal 0 to 9 hex code stored at code memory starting at location 0200H onward		
END	End of program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Sr. No.	HEX code	Display number/character
1	F9	
2	99	
3	90	

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions to ensure the achievement of identifies CO.

1. List the numbers for common cathode type display in which segment 'b' is activated.
2. Write assembly program to blink number 0 and 1 on seven segment display continuously.
3. Calculate values for common cathode display to display 0 to 9 numbers.

[Space for Answers]

...……..…

……

……...……

……

……

…….....

.....

.....

.....……

……..……

……..…

……..…

……..……

……..

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVIII References/Suggestions for further reading

1. https://en.wikipedia.org/wiki/Seven-segment_display
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Identifying components on developer kit	20%
4	Follow ethical practices.	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.29: Interfacing of LCD with 8051 microcontroller to display the alphabets and decimal numbers.

I Practical Significance

LCDs are used in a wide range of applications including Computers, televisions, instrument panels, calculator's displays. It also used for displaying the messages in a more interactive way to operate the system or displaying error messages etc. LCDs are interfaced with microcontrollers to display the given integer and character. This practical will help the students to develop skills to write assembly program for LCD.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Use microcontroller in various applications.

IV Laboratory Learning Outcome

Interface Liquid Crystal Display LCD with 8051.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

A liquid-crystal display (LCD) is a flat panel display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. An LCD is of two types depending upon how they made with either a passive matrix or an active-matrix display grid.

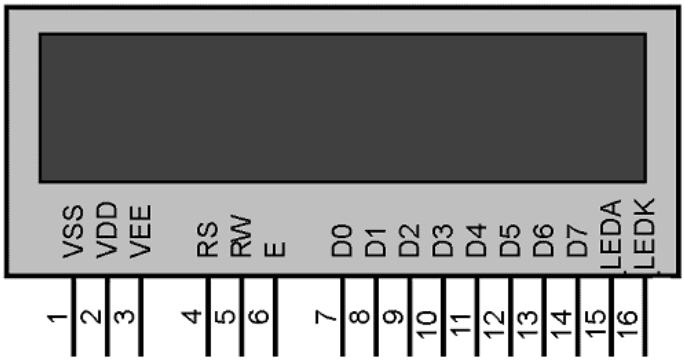


Fig 29.1 LCD Pins

Table 29.1 Control pin function

Rs -> Register select	RW -> Read/Write	EN -> Enable
Rs=1 -data register Rs=0 -command	RW=1 -reading from LCD. RW=0 -writing to LCD.	EN= high to low (Logic '1' delay Logic '0') for use LCD module.

Table 29.2 Operating parameters – Pin description of LCD module

Pin No	Function	Name	Pin No	Function	Name
1	Ground (0V)	Ground	**9**	8-bit data pins	DB2
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc	**10**	8-bit data pins	DB3
3	Contrast adjustment; through POT	V_{EE}	**11**	8-bit data pins	DB4
4	Selects command register	Register Select(RS)	**12**	8-bit data pins	DB5
5	Selects data register	Read/write(RW)	**13**	8-bit data pins	DB6
6	Sends data to data pins when a high to low pulse is given	Enable(E)	**14**	8-bit data pins	DB7
7	8-bit data pins	DB0	**15**	Backlight V_{CC} (5V)	LED+
8	8-bit data pins	DB1	**16**	Backlight Ground (0V)	LED-(GND)

Table 29.3 LCD commands

Code in Hex	Command to LCD instruction register	Code in Hex	Command to LCD instruction register
01	Clear Display Screen	0E	Display on, curser blinking
02	Return Home	0F	Display on, curser blinking
04	Decrement curser (Shift cursor to left)	10	Shift cursor Position to left
06	Increment curser (Shift cursor to right)	14	Shift curser Position to right
05	Shift display right	18	Shift the entire display to the left
07	Shift display left	1C	Shift the entire display to the right
08	Display off, curser off	80	Force curser to beginning to 1st line
0A	Display off, curser on	C0	Force cursor to beginning to 2nd line
0C	Display on, curser off	38	2 lines and 5*7 matrix (8 data lines)

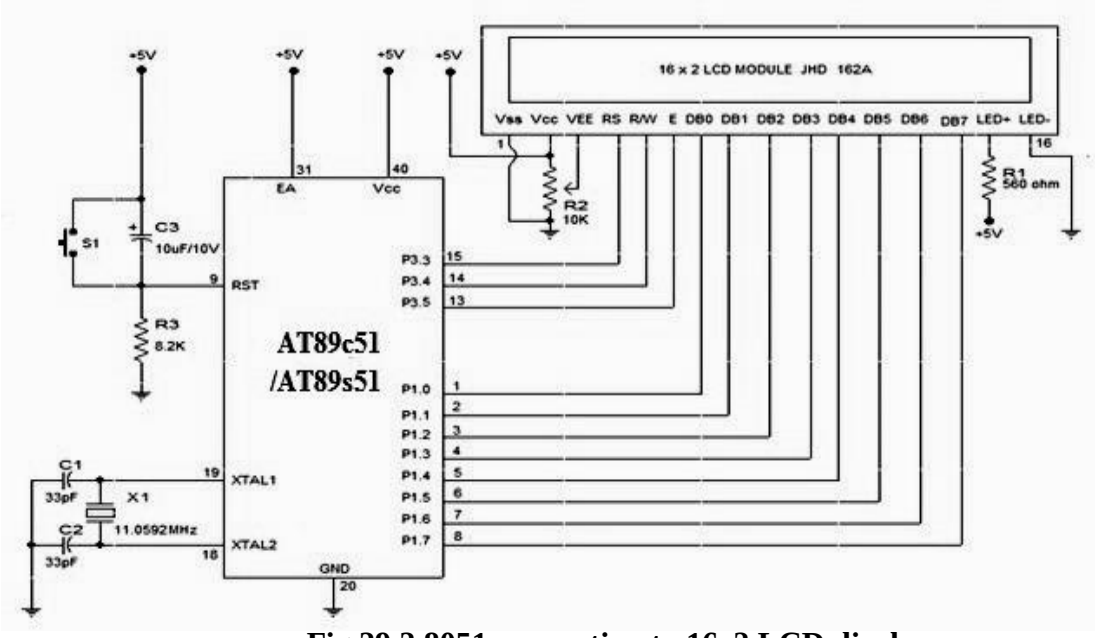


Fig 29.2 8051 connection to 16x2 LCD display

VII Actual Circuit diagram used in laboratory with related equipment rating.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1
3	Microcontroller kit	Single board systems with 8KRAM, ROM memory with battery backup, 16X2 LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler, RS-232, USB interfacing facility with built in power supply.	1
4	LCD trainer board	16x2 LCD suitable to interface with 8051 trainer kit	1

IX Precautions to be followed

1. LCD panel is made up of glass avoid applying strong pressure on to the surface of display area.
2. Ensure proper connection then only give electric supply to circuit.

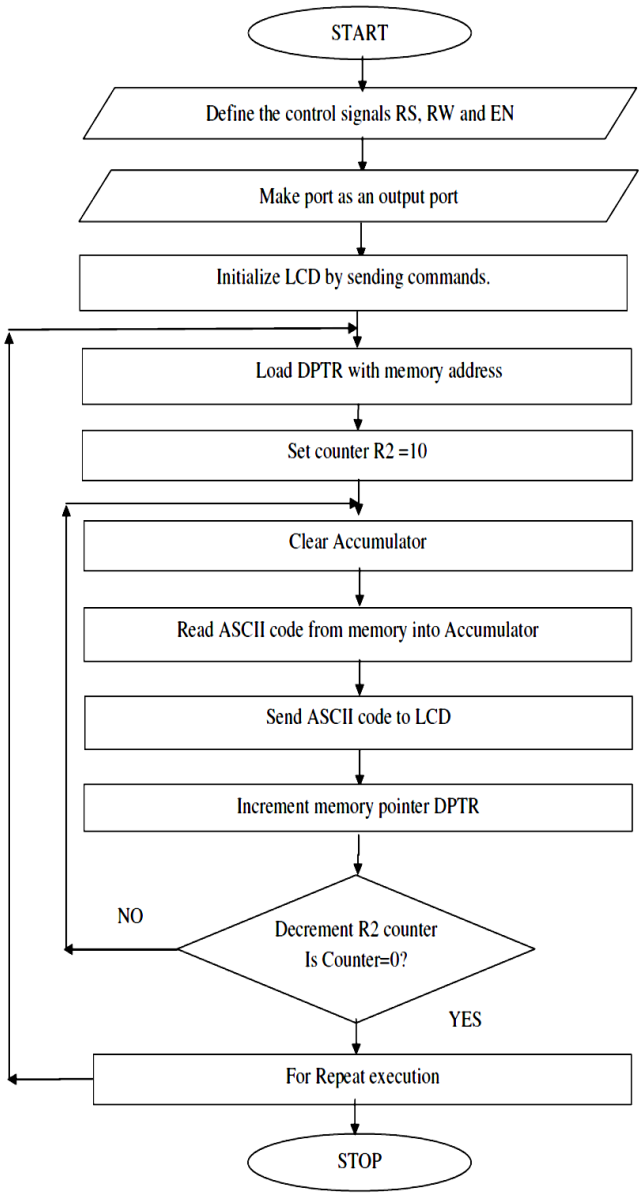
X Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Interface LCD display to microcontroller as per circuit diagram shown in Fig 29.2
8. Download hex code in EPROM/Flash memory of microcontroller
9. Observe output on LCD display and note it in observation Table

Sample program: Develop and execute ALP to display “uC AT89c51" on first line of LCD.	**Student activity: Develop and execute ALP to display “MSBTE” on first line of LCD.**
Algorithm: 1.Define control signals RS, RW and EN for LCD 2.Make LCD connected port as an output port. 3.Initialize LCD by sending commands. 4.Load DPTR with program memory address. 5.Set register as counter R2 =10 (decimal) for to display “uC AT89c51" 6.Clear Accumulator 7.Read ASCII from code memory into Accumulator. 8.Send code to output port where LCD is connected.	**Algorithm:**

9.Increment memory pointer.
10.Decrement R2 counter. Is count =0? NO- go to step 6.
11.For repeat operation go to step 4.
12.Stop

Flowchart:



Flowchart

Assembly language program

Instructions	Comments	Instructions	Comments
RS BIT P3.3	Replace a bit P3.3 address by a symbol RS		
RW BIT P3.4	Replace a bit P3.4 address by a symbol RW		
EN BIT P3.5	Replace a bit P3.5 address by a symbol EN		
ORG 0000H	Start of program		
MOV P1, #00H	Set P1 as output port		
LCALL LCD_INIT	Call LCD initialize subroutine		
MOV DPTR, #MSG	Load program memory address into DPTR		
MOV R2, #10	Set counter of 10		
DISP_MSG: CLR A	Clear A register		
MOVC A, @A+DPTR	Read data from memory into A register		
LCALL LCDDATA	Call LCDDATA subprogram		
INC DPTR	Increment pointer to next location		
DJNZ R2, DISP_MSG	Repeat loop for 10 times		
SJMP $	Wait		
LCD_INIT: MOV A, #38h	2 lines and 5×7 matrix (8-bit mode)		
ACALL LCDCMD	Call LCDCMD subprogram		
MOV A, #0Eh	Display on, cursor on		
ACALL LCDCMD	Call LCDCMD subprogram		
MOV A, #06h	Increment cursor (shift cursor to right)		
ACALL LCDCMD	Call LCDCMD subprogram		
MOV A, #01h	Clear display screen		
ACALL LCDCMD	Call LCDCMD subprogram		
MOV A, #80H	Force cursor to beginning to 1st line		
ACALL LCDCMD	Call LCDCMD subprogram		
RET	Return to main program		
LCDCMD:MOV P1, A	Send command to lcd		
CLR RS	Select command register		

CLR RW	Select write operation		
SETB EN	Set EN bit i.e. EN=1		
CLR EN	Latch command to lcd		
ACALL DELAY	Wait for sometime		
RET	Return to main program		
LCDDATA:MOV P1, A	Send data to lcd		
SETB RS	Select data register		
CLR RW	Select write operation		
SETB EN	Set EN bit i.e. EN=1		
CLR EN	Latch data to lcd		
ACALL DELAY	Wait for sometime		
RET	Return to main program		
DELAY: MOV R3, #50	Delay subroutine, Set counter of 50		
L2: MOV R4, #255	Set counter of 255		
L1: DJNZ R4, L1	Decrement R4 counter till it becomes zero		
DJNZ R3, L2	Decrement R3 counter till it becomes zero		
RET	Return to main program		
ORG 0050H	Display message stored at code memory starting at location 0050H onward		
MSG:DB "uC AT89c51"	Define data byte to code memory		
END	End of program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Sr. No.	ASCII Symbol	LCD memory location
1	A	
2	T	
3	8	
4	9	

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions to ensure the achievement of identifies CO.

1. Write down cost of LCD module for at least 3 different Manufacturer with specifications.
2. Write assembly program for checking busy flag in LCD display.
3. Write appropriate command to shift entire data to left.

[Space for Answers]

...……..…

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://en.wikipedia.org/wiki/Liquid-crystal_display
2. http://www.keil.com
3. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Identifying components on developer kit	20%
4	Follow ethical practices.	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.30: Interfacing of stepper motor with 8051 microcontroller and write ALP to rotate stepper motor in clockwise and anti-clockwise direction at given angles.

I Practical Significance

Different field applications require precise positioning, repeatability of movement in clockwise and anticlockwise direction with good accuracy. Stepper motors are controlled by microcontrollers in such areas like in computer peripherals, Business machines, process control and for making robots. This practical will help the students to develop skills to interface stepper motor to 8051 and rotate in clockwise and anticlockwise direction.

II Industry/Employer Expected Outcome

Use microcontroller based systems for various industrial applications.

III Course Level Learning Outcome

Use microcontroller in various applications.

IV Laboratory Learning Outcome

Interface stepper motor with 8051.

V Relevant Affective Domain related outcome

Follow ethical practices.

VI Relevant Theoretical Background

Stepper motors convert electrical energy into precise mechanical motion. These motors rotate a specific incremental distance per each step. The number of steps executed controls the degree of rotation of the motor's

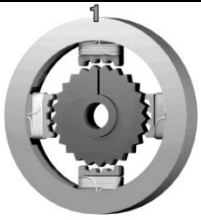
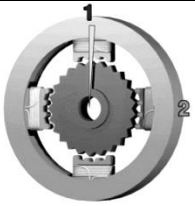
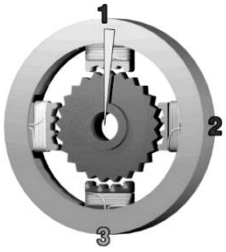
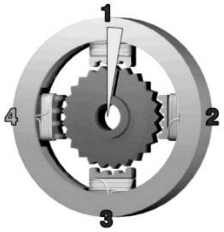
Table 30.1 Stepper Motor Step Angles

Step Angle	Steps per revolution
0.72	500
1.8	200
2.0	180
2.5	144
5	72

Table 30.2 Two Coil Excitation Full Step Sequence

P2.3	P2.2	P2.1	P2.0	Port Pin	
Stator Windings of Stepper Motor					
A	B	C	D	Hex code	Direction
1	0	0	1	09 H	CW ↓
1	1	0	0	0C H	
0	1	1	0	06 H	
0	0	1	1	03 H	↑ CCW

Table 30.3 Stepper motor working

	
Frame 1: The top electromagnet (1) is turned on, attracting the nearest teeth of the gear-shaped iron rotor. With the teeth aligned to electromagnet 1, they will be slightly offset from right electromagnet (2).	**Frame 2:** The top electromagnet (1) is turned off, and the right electromagnet (2) is energized, pulling the teeth into alignment with it. This results in a rotation of 1.8° in this example.
	
Frame 3: The bottom electromagnet (3) is energized; another 1.8° rotation occurs.	**Frame 4:** The left electromagnet (4) is energized, rotating again by 1.8°. When the top electromagnet (1) is again enabled, the rotor will have rotated by one tooth position; since there are 50 teeth, it will take 200 steps to make a full rotation in this example.

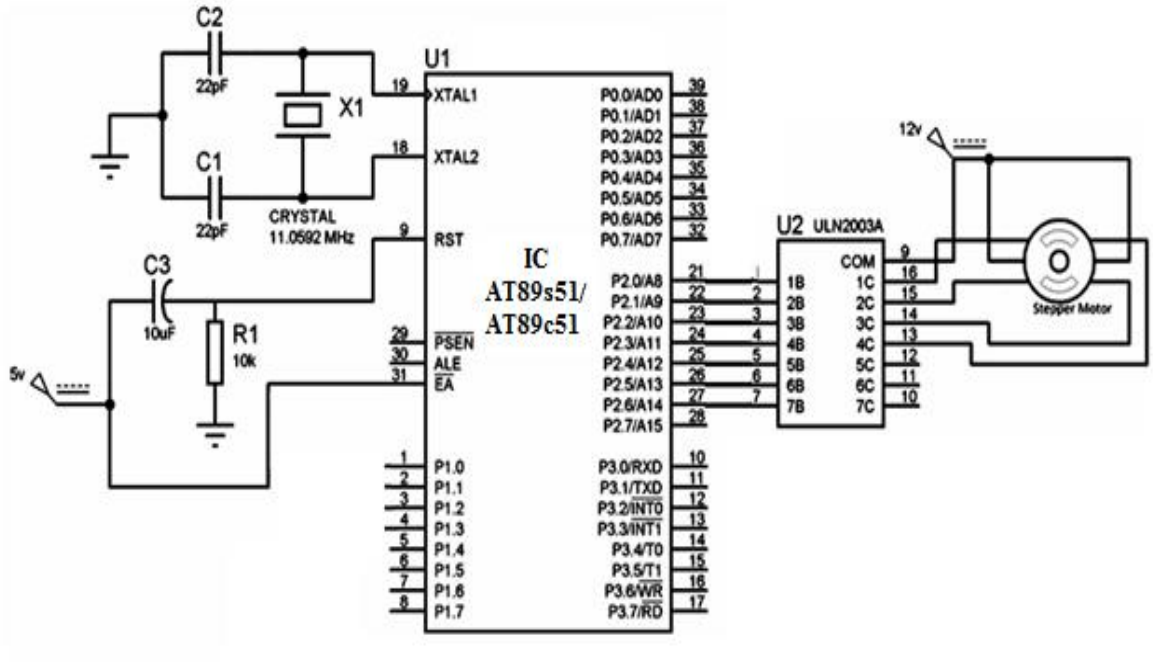


Fig. 30.1 8051 connection to stepper motor

VII Actual Circuit diagram used in laboratory with related equipment rating.

VIII Required Resources/apparatus/equipment with specifications

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Desktop PC	Core i3/i5 Processor, 32/64-bit windows operating system, 2GB/Higher RAM	1
2	Keil or any relevant open-source IDE, simulation and program downloading software.	Compatible with Desktop PC	1
3	Microcontroller kit	Single board systems with 8KRAM, ROM memory with battery backup, 16X2 LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler, RS-232, USB interfacing facility with built in power supply.	1
4	Stepper Motor	1.8-degree step angle, 50/100 RPM with driver circuitry	1

IX Precautions to be followed

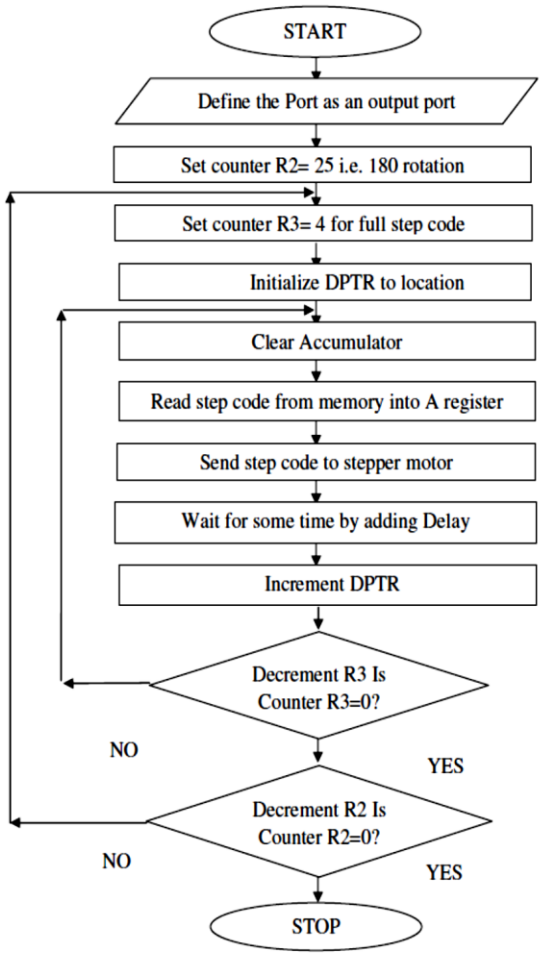
1. Use always driver circuit while interfacing stepper motor to microcontroller.
2. Check rules / syntax of assembly programming.

X Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Download hex code in EPROM/Flash memory of the microcontroller.
8. Interface stepper motor to microcontroller as per circuit diagram shown in fig 30.1
9. Observe rotation of stepper motor and record in observation Table.

Sample program: Develop assembly language program to rotate stepper motor in clockwise direction by $180^{0.}$ Assume step angle 1.8°	**Student activity: Develop assembly program for stepper motor to rotate 90° in anticlockwise direction using rotate instruction. Assume step angle 1.8°**
Algorithm: 1. Make the Port used to Interface stepper motor as an output port. 2. Set register as counter R2 =25 for 100 steps i.e.180° rotation. 3. Set register as counter R3= 4 for full step code 4. Initialize pointer to table which is in code memory i.e. DPTR. 5. Clear accumulator. 6. Read data from code memory. 7. Send code to stepper motor. 8. Increment DPTR to access next memory location code. 9. Decrement R3 and check for zero. Is counter R3=0? NO- go to step 5 else go to next. 10. Decrement R2 and check for zero. Is counter R2=0? NO- go to step 3 else go to next. 11. Stop.	**Algorithm:**

Flowchart:



Flowchart

Assembly language program

Instructions	Comments	Instructions	Comments
ORG 0000H	Start of program		
MOV P1, #00H	Define output port		
MOV R2, #25	Set counter of 25 for 180-degree rotations		
UP1:MOV R3, #4	Set counter of 4 for full step code sequence		
MOV DPTR, #TABLE	Load address of program memory into DPTR		
UP: CLR A	Clear accumulator		
MOVC A, @A+DPTR	Read step code from memory into accumulator		
MOV P1, A	Send step code to port		
ACALL DELAY	Add delay		

INC DPTR	Increment memory pointer to read next step code		
DJNZ R3, UP	Decrement counter and jump to memory location labeled as UP if not equal to zero.		
DJNZ R2, UP1	Decrement counter and jump to memory location labeled as UP1 if not equal to zero.		
SJMP $	Wait		
DELAY:MOV R4, #25	Delay Subroutine, Set counter of 25		
L3:MOV R5, #100	Set counter of 100		
L2:MOV R6, #100	Set counter of 100		
L1: DJNZ R6, L1	Decrement R6 counter till it becomes zero		
DJNZ R5, L2	Decrement R5 counter till it becomes zero		
DJNZ R4, L3	Decrement R4 counter till it becomes zero		
RET	Return to main program		
ORG 0050H	Origin of TABLE		
TABLE:DB 09H, 0CH, 06H, 03H	Step code stored at code memory starting at location 0050H onward.		
END	End of program		

XI Resources used

Sr. No.	Name of Resource	Specifications	Quantity

XII Actual Procedure

……..

……

..……

……..

XIII Observation Table

Step Code	Port Pin	Status - logic 1 (+5V) / logic 0 (0V)
9H	P2.4 - P2.0	
CH	P2.4 - P2.0	
6H	P2.4 - P2.0	
3H	P2.4 - P2.0	

XIV Result(s)

……..

……

..……

……..

XV Interpretation of results

……..

.....……

……..

…….....

XVI Conclusion and recommendation

……

……

……

XVII Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions to ensure the achievement of identifies CO.

1. Write any four specifications of different stepper motors available in the market.

2. Calculate no of steps to move stepper motor 64 degree in clockwise direction if step angle is 2 degrees, steps per revolution 180 and number of rotor teeth 45.
3. If a motor takes 90 steps to make one complete revolution, calculate the step angle for this motor?
4. Write any two applications of stepper motor.

[Space for Answers]

……

……

……

……

……

……

……

……

……

……

……

XVIII References/Suggestions for further reading

1. https://en.wikipedia.org/wiki/Stepper_motor.
2. https://en.wikipedia.org/wiki/Intel_MCS-51

XIX Assessment Scheme

Performance Indicators		Weightage
Process Related: 15 Marks		**60 %**
1	Use of IDE tools for programming	10%
2	Coding and Debugging ability	20%
3	Identifying components on developer kit	20%
4	Follow ethical practices.	10%
Product Related: 10 Marks		**40%**
5	Correctness of algorithm/ Flow chart	20%
6	Relevance of output of the problem definition	15%
7	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	