I

Name ______________________

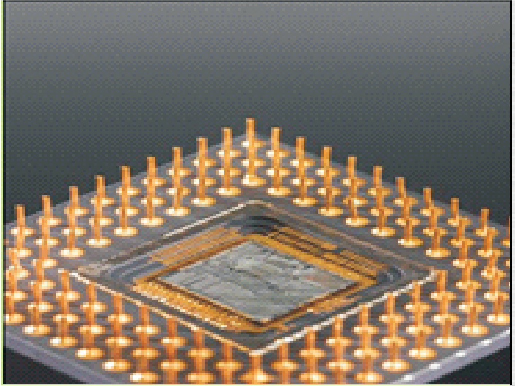
Roll No. ______________ Year 20______ 20______

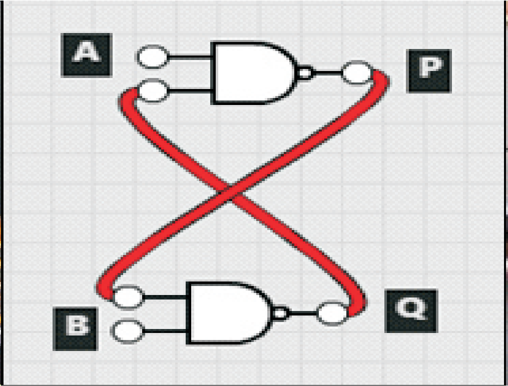
Exam Seat No. ______________________

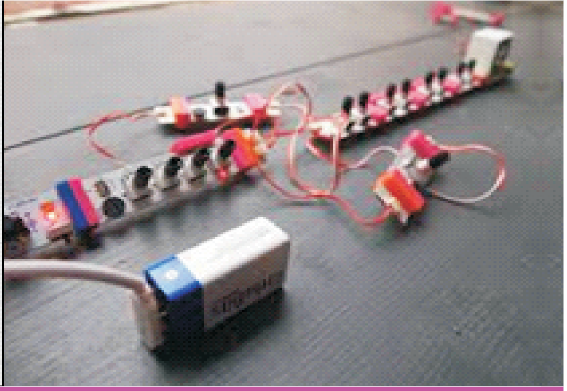
COMPUTER GROUP | SEMESTER - III | DIPLOMA IN ENGINEERING AND TECHNOLOGY

A LABORATORY MANUAL FOR

DIGITAL TECHNIQUES & MICROPROCESSOR (22323)







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 programmes.

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 program.
- Technical skills do need support of life skills.
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.

A Laboratory Manual

for

Digital Techniques and Microprocessor

(22323)

Semester-III

(IF)

Maharashtra State
Board of Technical Education, Mumbai

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

Maharashtra State Board of Technical Education,
(Autonomous) (ISO:9001:2015) (ISO/IEC 27001 : 2013)
4th Floor, Government Polytechnic Building, 49, Kherwadi,
Bandra (East), Mumbai - 400051.
(Printed on June, 2018)

MAHARASHTRA STATE

BOARD OF TECHNICAL EDUCATION

Certificate

This is to certify that Mr. / Ms. ………………………………………..……… Roll No. …………………., of Third Semester of Diploma in ……………………..…………………………………………………… of Institute, ……………..…………………………………………………………..… (Code: …………...) has completed the term work satisfactorily in course **Digital Techniques and Microprocessor (22323)** for the academic year 20…….. to 20…….. as prescribed in the curriculum.

Place: …………… Enrollment No:.……………

Date: …………… Exam. Seat No: ……………

Subject Teacher **Head of the Department** **Principal**

Seal of
Institution

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 'I' Scheme curricula for engineering diploma programmes with outcome-based education as the focus and accordingly, 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 to realize that every minute of the laboratory time need 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 'I' 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 pre-determined 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 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.

It is essential to know fundamentals of digital electronics to understand the concept of microprocessor and its application. Microprocessor benefits to meet challenges of growing applications of advanced microprocessor based technologies hence students are expected to be conversant with components of microprocessors and microprocessor based programming. This course is designed to help the students to design logic circuits and to understand the architecture of 8086 microprocessor. The course also enables students to develop assembly language programs using instruction set of 8086 Microprocessor.

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) to be achieved through Practical of this Course:-

PO1. **Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.

PO 2. **Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.

PO 3. **Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.

PO 4. **Engineering tools:** Apply appropriate Information Technology related techniques/ tools with an understanding of the limitations.

PO 5. **The engineer and society:** Assess societal, health, safety and legal issues and the consequent responsibilities relevant to practice in the field of Information technology.

PO 6. **Environment and sustainability:** Apply Information Technology related engineering solutions for sustainable development practices in environmental contexts.

PO 7. **Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of practice in the field of Information Technology.

PO 8. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

PO 9. **Communication:** Communicate effectively in oral and written form.

PO 10. **Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

Practical- Course Outcome matrix

Course Outcomes (COs):

a. Test the Digital Systems, Logic Families and logic gates.
b. Construct combinational logical circuit.
c. Construct sequential logical circuit.
d. Use registers and instructions of 8086.
e. Develop assembly language programs using 8086.

S. No.	Practical Outcome	CO a.	CO b.	CO c.	CO d.	CO e.
1	**To study and verify the truth table of logic gates** Verify truth tables of basic logic gates using Transistor-Transistor Logic (TTL) Integrated Circuits (ICs).	√	-	-	-	-
2	**To study and verify the truth table of Universal logic gates** Check truth tables of universal logic gates (NAND and NOR) using Transistor-Transistor Logic (TTL) Integrated Circuits (ICs).	√	-	-	-	-
3	**Understand and Verify De Morgan's Theorem** **Check** De Morgan's theorem using ICs	√	-	-	-	-
4	**Convert given expression to Sum of Product (SOP) form using basic logic gates.**	√	√	-	-	-
5	**Convert given expression to Product of Sum (POS) form using basic logic gates.**	√	√	-	-	-
6	**Implement Combinational Circuit using Multiplexer** Verify Truth Table of half Adder & Half Subtractor using 4:1 mux 74153 IC.	√	√	-	-	-
7	**Construct S-R, J-K, D and T flip-flop and verify their truth tables.**	√	-	√	√	-
8	**Evaluate Arithmetic Operations** Write and execute an Assembly Language Program(ALP) to add / subtract two 8 bit and 16 bit numbers with the help of programming tools and any simulator	-	-	-	√	√
9	**Perform Sum Of Series Of Numbers** Write and execute an ALP to find sum of series of 8 bit and 16 bit numbers.	-	-	-	√	√
10	**Use Of Assembly Language In Signed And Unsigned Multiplication** Develop an ALP to multiply two 8 bit and 16 Bit numbers. Unsigned/ signed numbers.	-	-	-	√	√
11	**Use Of Assembly Language In Signed And Unsigned Division** Develop an ALP to divide two 8 bit and 16 bit Numbers. Unsigned/ signed numbers.	-	-	-	√	√

12	**Binary Coded Decimal Addition / Subtraction Of BCD Numbers** Write an ALP to add / Subtract two BCD Numbers.	-	-	-	√	√
13	**Binary Coded Decimal Multiplication/ Division Of BCD Numbers** Write an ALP to multiply / Divide two BCD numbers	-	-	-	√	√
14	**Find Smallest Number From An Array.** Develop an ALP to find smallest number from array of n numbers.	-	-	-	√	√
15	**Find Largest Number From An Array** Develop an ALP to find largest number from array of n numbers.	-	-	-	√	√
16	**Program To Perform Block Transfer** Write an ALP to perform block transfer from One memory location to another.	-	-	-	√	√

List of Industry Relevant Skills

The following industry relevant skills of the competency are expected to be developed in you by undertaking the practicals of this laboratory manual.

Digital Techniques and Microprocessor:

1. Analyze problem definition
2. Build digital circuits
3. Develop Assembly programs for Real life applications.
4. Ability to solve application level problems.

Guidelines to Teachers

1. Faculty should provide the guideline with demonstration of practical to the students with all features.
2. Faculty shall explain prior concepts to the students before starting of each experiment.
3. Faculty Involve students in performance of each experiment.
4. Faculty should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
5. Faculty should give opportunity to students for hands on experience after the demonstration.
6. Faculty is expected to share the skills and competencies to be developed in the students.
7. Faculty may provide additional knowledge and skills to the students even though not covered in the manual but are expected the students by the industry.
8. Faculty is expected to encourage students to refer references mentioned in the manual for better understanding

Instructions for Students

1. Students shall listen carefully the lecture given by Faculty about subject, curriculum, learning structure, skills to be developed.
2. Students shall organize the work in the group of two or three and make record of all observation (wherever applicable).
3. Students shall develop maintenance skill as expected by industries.
4. Student shall attempt to develop related hands-on skills and gain confidence.
5. Student shall develop the habits of evolving more ideas, innovations, skills etc. those included in scope of manual.
6. Student shall refer references mentioned in the manual for better understanding.
7. Student shall develop habit to submit the practical related work on time.
8. Student shall Attach /paste separate papers wherever necessary.

Content Page

List of Practicals and Progressive Assessment Sheet

S. No	Practical Outcome	Page No.	Date of performance	Date of submission	Assessment marks (25)	Dated sign. of teacher	Remarks (if any)
10.	**Use Of Assembly Language In Signed And Unsigned Multiplication** Develop an ALP to multiply two 8 bit and 16 bit numbers. Unsigned/ signed numbers.	81					
11.	**Use Of Assembly Language In Signed And Unsigned Division** Develop an ALP to divide two 8 bit and 16 bit numbers. Unsigned/ signed numbers.	91					
12.	**Binary Coded Decimal Addition / Subtraction Of BCD Numbers** Write an ALP to add / Subtract two BCD numbers.	101					
13.	**Binary Coded Decimal Multiplication/ Division Of BCD Numbers** Write an ALP to multiply / Divide two BCD numbers	112					
14.	**Find Smallest Number From An Array.** Develop an ALP to find smallest number from array of n numbers.	121					
15.	**Find Largest Number From An Array** Develop an ALP to find largest number from array of n numbers.	133					
16.	**Program To Perform Block Transfer** Write an ALP to perform block transfer from one memory location to another.	144					
Total							

- **To be transferred to Proforma of CIAAN-2017.**

Practical No. 1: To study and verify the truth table of logic gates

I Practical Significance

Implementation of logic circuits with a minimum of logic gates where basic logic gates are the building blocks of more complex logic circuits. These basic logic gates are implemented as small-scale integrated circuits (SSICs) or as part of more complex medium scale (MSI) or very large-scale (VLSI) integrated circuits. Digital IC gates are classified not only by their logic operation, but also the specific logic-circuit family to which they belong. Each logic family has its own basic electronic circuit upon which more complex digital circuits and functions are developed with the help of various logic families.

II Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2: Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 3. Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
4. **PO 4. Engineering tools:** Apply appropriate Information Technology related techniques/ tools with an understanding of the limitations.
5. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III Competency and Practical Skills

This practical is expected to develop the following skills in students:

'Build/ test digital logic circuits consist of digital ICs.

i. Identify pin configuration of logic gate IC's.

ii. Test the functionality of the logic gates.

IV Relevant Course Outcome(s)

i) Test the Digital Systems, Logic Families and logic gates.

V Practical Outcome

Verify truth tables of basic logic gates using Transistor-Transistor Logic (TTL) Integrated Circuits (ICs)

VI Relevant Affective domain related Outcome(s)

1. Handle IC and equipment carefully.
2. Follow safe practices.

VII Minimum Theoretical Background

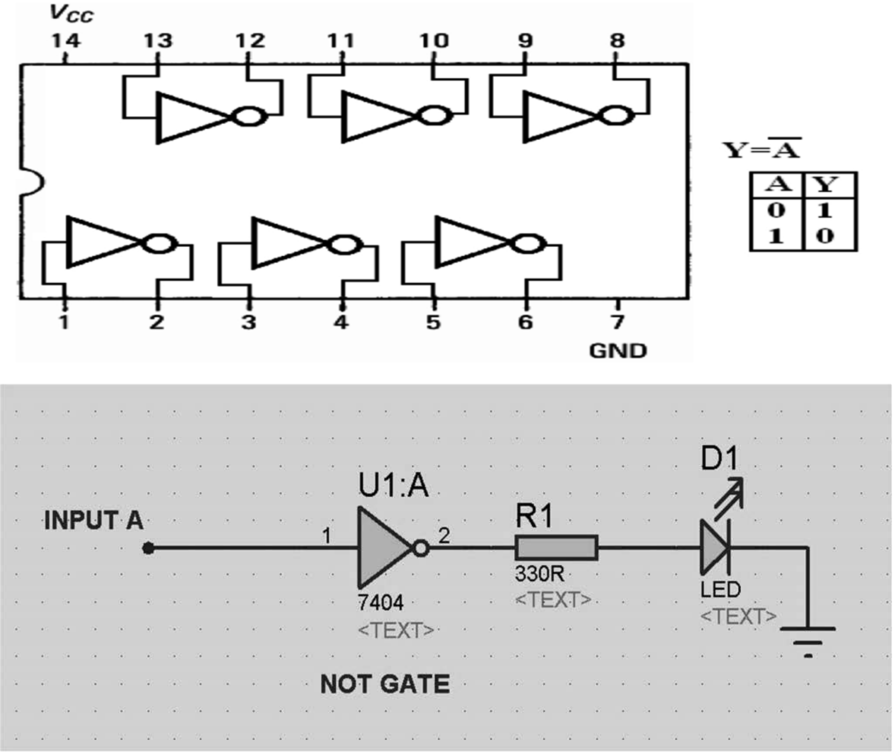
A logic gate is an electronic circuit which makes logical decisions. It has only one output and one or more inputs. In digital logic design only two voltage levels or states are allowed and these states are generally referred to as Logic "1"or High represented by +5V and Logic "0", or Low represented by 0V. Digital systems are said to be constructed by using logic gates like AND, OR, NOT, and EXOR gates. These gates are verified using Truth tables which help to understand the behavior of logic gates.

Classification of Logic Gates:

Logic Gates

Basic Gates	Universal Gates	Special Purpose Gates
NOT, **AND** & **OR** Gate	**NAND** & NOR Gate	**EX-OR** & **EX-NOR** Gate

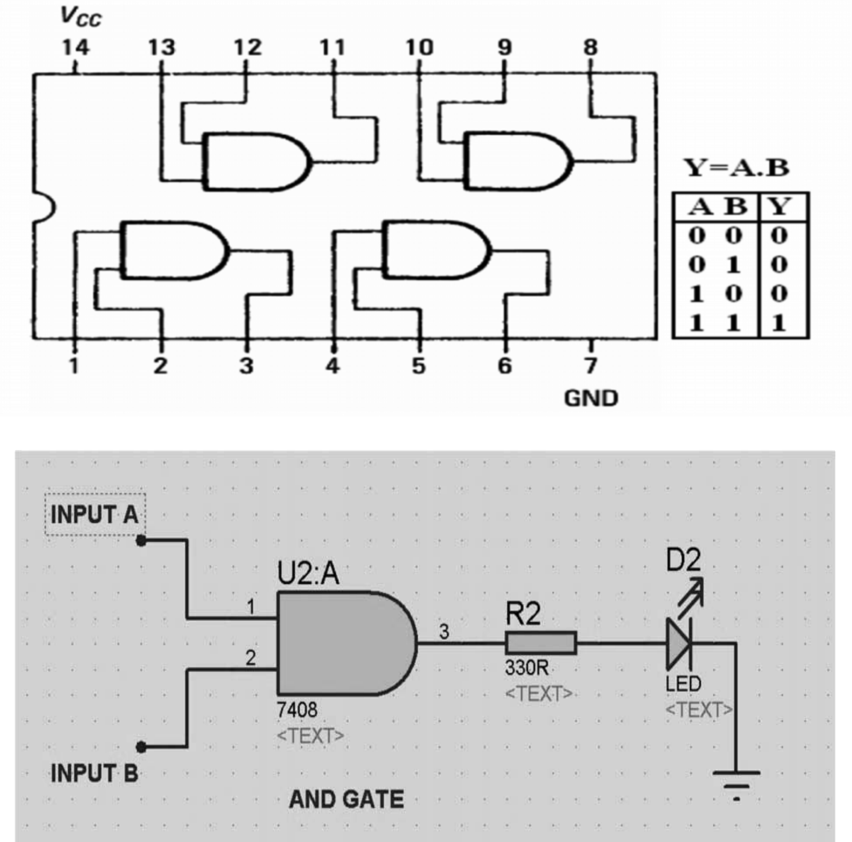
VIII. Practical set-up / Circuit diagram



$Y=\overline{A}$

A	Y
0	1
1	0

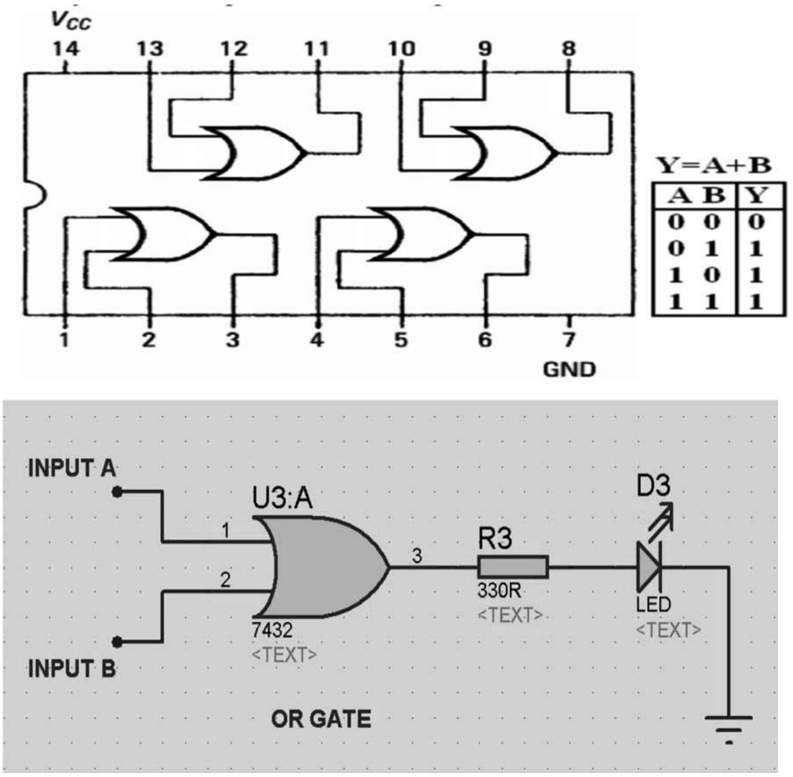
Figure 1.1: NOT Gate IC 7404



Y=A.B

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

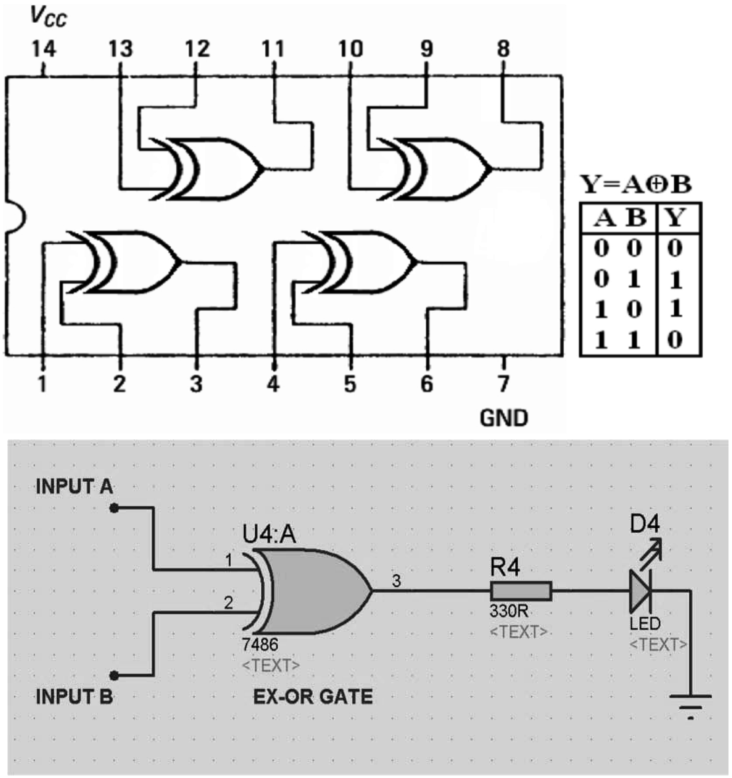
Figure 1.1: AND gate IC 7408



Y=A+B

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1.3: OR Gate IC 7432



Y=A⊕B

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Figure 1.4: EX-OR Gate IC 7486

IX. Resources Required

S. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboards	General Purpose Breadboards	1
3	DC power supply	+5 V Fixed power supply	1
4	IC	7486, 7404, 7432, 7408	1 Each
5	LED	Red color 5 mm	1
6	Connecting wires	Single strand 0.6 mm Teflon coating	LS
7	Digital IC tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS Ic’s	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply (0-5V/500mA)

XI Procedure

1. Make the connection as per circuit diagram and give supply voltage to relevant pin
2. Connect the inputs from source to logic gates as per logic level.
3. Observe the output on LED for each combination of input as per truth table.
4. Verify the truth table.
5. Repeat the process for all other logic gates.

XII Resources Used

S. No.	Name of Resource	Broad Specifications		Quantity	Remarks (If any)
		Make	Details		
1.					
2.					
3.					

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

XIV Precautions Followed

....

....

....

XV Observations and Calculations (use blank sheet provided if space not sufficient)

Inputs		7404 (NOT)		7408(AND)		7432(OR)		7486(EX-OR)	
A	B	LED Status (ON/OFF)	Output voltage	LED Status (ON/OFF)	Output voltage	LED Status (ON/OFF)	Output voltage	LED Status (ON/OFF)	Output voltage
0(0V)	0(0V)								
0(0V)	1(5V)								
1(5V)	0(0V)								
1(5V)	1(5V)								

XVI Results

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

....

....

....

XIX Practical Related Questions

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

1. Write down voltage at logic level 0 and 1.
2. List the function of pin 7,14of IC 7432.
3. What will happen if pin number 14 is connected to ground and pin number 7 is connected to VCC?
4. List number of NOT gates used in IC 7404.
5. Write down the name of manufacturer of Digital IC used in practical.
6. State the need of resister connected in series with LED .Write down the value of resister.

(**Space for Answers**)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

XX References / Suggestions for further reading

1. http://www.electronics-tutorials.ws/logic/logic_1.html
2. https://academo.org/demos/logic-gate-simulator
3. https://www.youtube.com/watch?v=AT_GjUjNFpo
4. https://www.youtube.com/watch?v=EBlgoycFNJ8
5. https://www.youtube.com/watch?v=LBuLmC0chyQ
6. https://www.youtube.com/watch?v=WGYEpZQnRE8
7. http://www.ti.com/lit/ds/symlink/sn74ls00.pdf

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Names of Student Team Members

1. …………………………
2. …………………………
3. …………………………
4. …………………………

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

Practical No. 2: To study and verify the truth table of universal logic gates

I Practical Significance

NAND and NOR universal gates can implement any Boolean function which are economical and easier to fabricate used in all IC digital logic families.

II Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 4: Engineering tools:** Apply appropriate Information Technology related techniques/ tools with an understanding of the limitations.
4. **PO 10: Life-long learning*:*** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III Competency and Practical Skills

This practical is expect to develop the following skills in students

'Build/ test digital logic circuits consist of digital ICs.'

i. Identify pin configuration of logic gate IC's.
ii. Test the functionality of the logic gates.

IV Relevant Course Outcome(s)

i. Test the functionality of the logic gates.

V Practical Outcome

Verify truth tables of Universal logic gates using Transistor-Transistor Logic (TTL) Integrated Circuits (ICs)

VI Relevant Affective domain related Outcome(s)

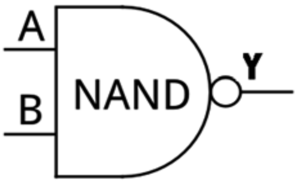
1. Handle IC and equipment carefully.
2. Follow safe practices.

VII Minimum Theoretical Background

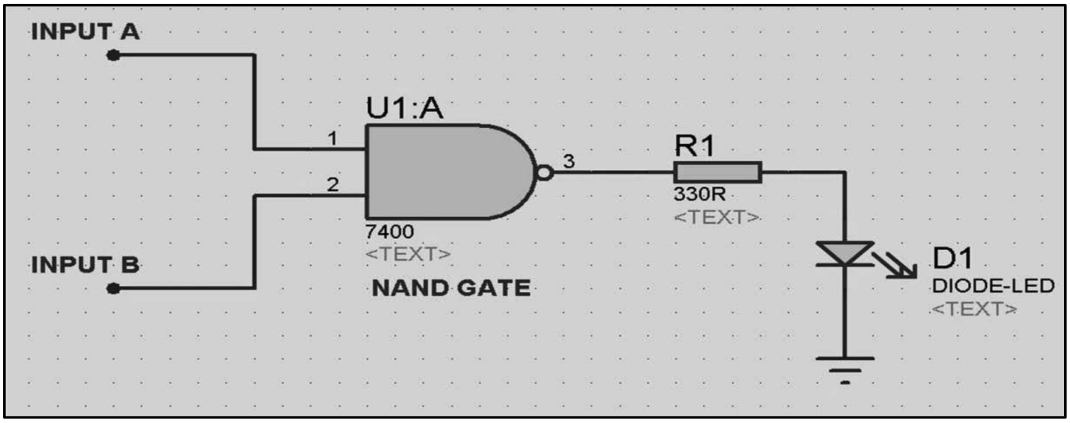
NOR and NAND gates have the property that they individually can be used to hardware-implement a logic circuit corresponding to any given Boolean expression. That is, it is possible to use either only NAND gates or only NOR gates to implement any Boolean expression. NAND gates are Universal gate. The Basic gates AND, OR, NOT can be realized from it. The Boolean equations and logic diagrams are as follows.

VIII Practical set-up / Circuit diagram: -

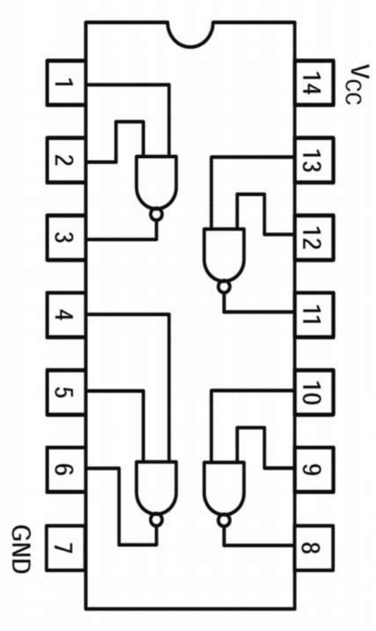
NAND gate



INPUT		OUTPUT
A	**B**	**Y=A.B**
0	**0**	**1**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**



Symbol

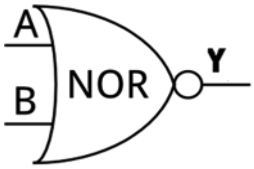


Truth Table
Characteristics of
7400 Series Families

Parameter	7400	Units
(V_{DD} = 5 V)		
V_{IH} (min)	3.5	V
V_{OH} (min)	4.5	V
V_{IL} (max)	1.5	V
V_{OL} (max)	0.5	V
I_{IH} (max)	1	μA
I_{IL} (max)	1	μA
I_{OH} (max)	0.4	mA
I_{OL} (max)	0.4	mA

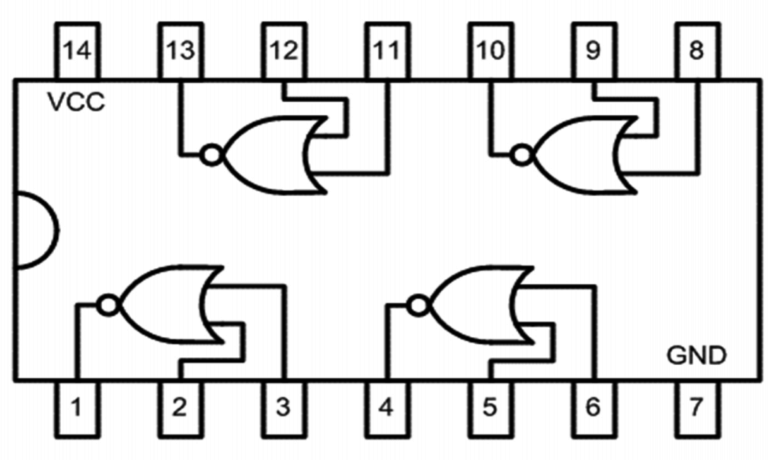
Figure 2.1 NAND Gate IC 7400

NOR gate



INPUT		OUTPUT
A	B	Y=$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Symbol **Truth Table**



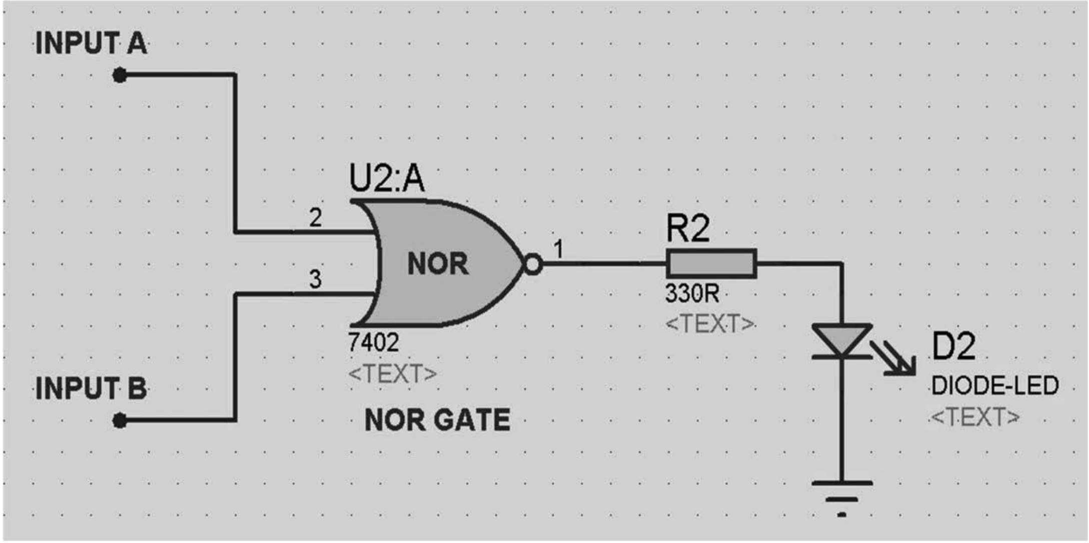


Figure 1.2 NOR gate IC 7402

IX Resources Required

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboard	General Purpose Breadboard	1
3	DC power supply	+5 V/500MA Regulated power supply	1
4	IC	7400, 7402	1 Each
5	LED, Resisters	Red color 5 mm ,330R	1 Each
6	Connecting wires	Single strand 0.6 mm Teflon coating	LS
7	Digital IC tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS Ic's	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply

XI Procedure

1. Make the connection as per circuit diagram and give supply voltage to relevant pin
2. Connect the inputs from source to logic gates as per logic level.
3. Observe the output on LED for each combination of input as per truth table.
4. Verify the truth table.
5. Repeat the process for other universal logic gate.

XII Resources Used

Sr. No.	Name of Resource	Broad Specifications		Quantity	Remarks (If any)
		Make	Details		
1.					
2.					
3.					
4.					
5.					

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

XIV Precautions Followed

....

....

....

....

XV Observations and Calculations (use blank sheet provided if space not sufficient)

Inputs		7400(NAND)		7402(NOR)	
A	B	LED Status (ON/OFF)	Output voltage	LED Status (ON/OFF)	Output voltage
0(0V)	0(0V)				
0(0V)	1(5V)				
1(5V)	0(0V)				
1(5V)	1(5V)				

XVI Results

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

....

....

....

XIX Practical Related Questions

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

1. Construct basic gates NOT, AND, OR using NAND gate. Write necessary outputs.
2. List number of NOR gates used in IC 7402.
3. Explain what is truth table?
4. How many NAND gates are there in a single 7400 NAND gate IC?
5. Explain why NAND-NAND realization is preferred over AND-OR realization.
6. Design AND, OR gate using NOR gate only.
7. Minimum voltage needed by Gate IC.

[Space for answer]

....

....

....

....

....

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

....

....

....

....

....

XX References / Suggestions for further reading

1. https://www.youtube.com/watch?v=lFt_xWcOzXY //NAND GATE DEMO
2. https://www.youtube.com/watch?v=146DbuTCzGQ //NOR GATE DEMO
3. https://en.wikipedia.org/wiki/List_of_7400_series_integrated_circuits

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Names of Student Team Members

1. …………………………
2. …………………………
3. …………………………
4. …………………………

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

Practical No. 3: Understand and Verify De Morgan's Theorem

I Practical Significance

Logic gates are the basis for building more complex logic circuits that are constructed using various combinations of gates which is known as Combinational Logic. Combinational logic requires the use of two or more gates to form a useful, complex function. These complex functions usually begin as a Boolean Equation and the logic circuit may be implemented directly from this equation. **De Morgan's Theorem** Simplifies the Boolean equations.

II Relevant Program Outcomes (POs)

1. **PO 1. Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2. Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 3. Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
4. **PO 4.Engineering tools:** Apply appropriate Information Technology related techniques/ tools with an understanding of the limitations.
5. **PO 10: Life-long learning:** Engage in independent and life -long learning along with the technological changes in the IT and allied industry.

III Competency and Skills

This practical is expected to develop the following skills in students:

'**Build/ test digital logic circuits consist of digital ICs.**

i. Test the Digital Systems, Logic Families and logic gates

IV Relevant Course Outcome(s)

Test the Digital Systems, Logic Families and logic gates

V Practical Outcome

Check De-Morgan's Theorem using ICs

VI Relevant Affective domain related Outcome(s):

- Follow Safety Practices
- Handle IC and equipment carefully

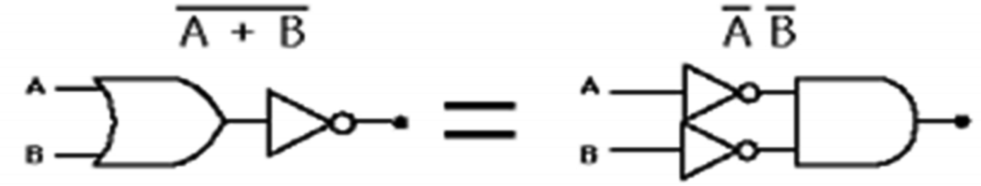
VII Minimum Theoretical Background

- **De-Morgan's first theorem:**

It states that the complement of a sum is equal to the product of complements.

1st Theorem: NOR Gate ⟹ Negative AND Gate (Babbled AND Gate)

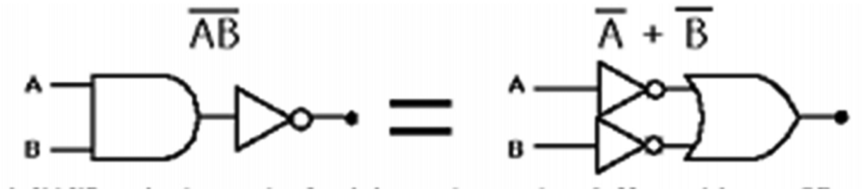
$$\boxed{\overline{A + B} = \bar{A}.\bar{B}}$$



- **De-Morgan's second theorem:**
  It states that the complement of a product is equal to the sum of complements.

2nd Theorem: NAND Gate ⟹ Negative OR Gate (Babbled OR Gate)

$$\boxed{\overline{A.B} = \bar{A} + \bar{B}}$$



VIII Circuit diagram

- **De Morgan's First Theorem:**

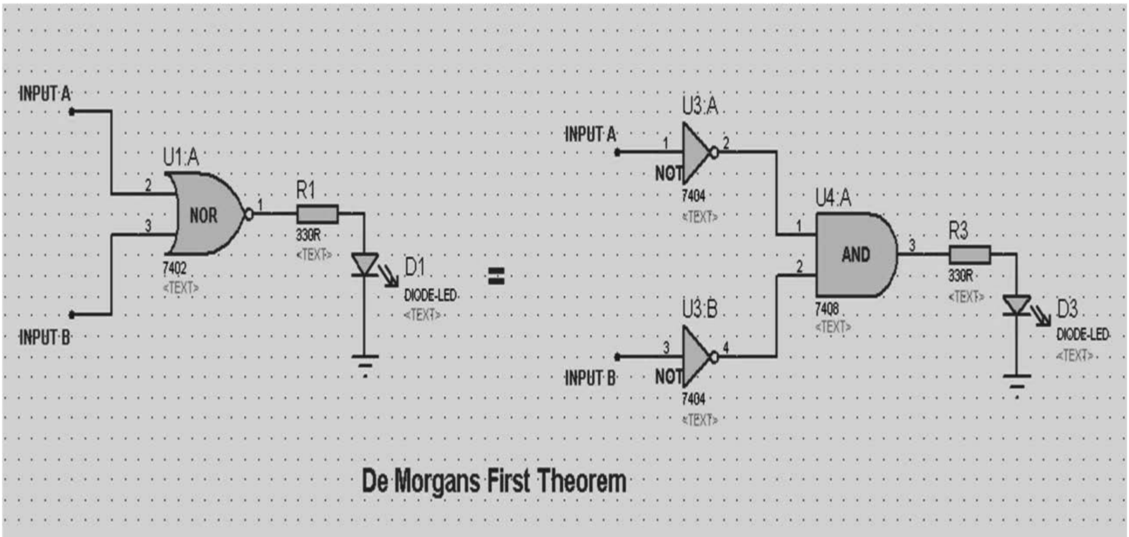


Figure 3.1 Circuit Details

- **De-Morgan's second theorem**

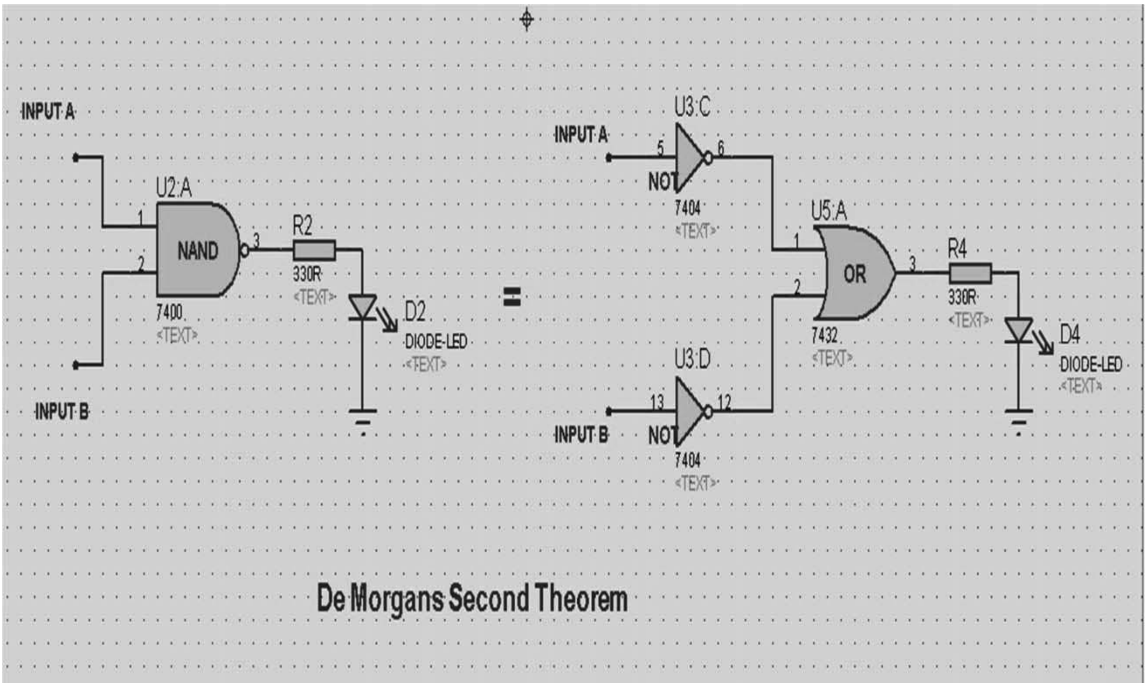


Figure 3.2 Circuit Details

IX Resources Required

S. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboard	General Purpose Breadboard	1
3	DC power supply	+5 V Fixed power supply	1
4	IC	7400,7402,7404,7408, 7432	1 Each
5	LED, Resisters	Red color 5 mm ,330R	1 Each
6	Connecting wires	Single strand 0.6 mm Teflon coating	LS
7	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS Ic's	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply (0-5V/500mA)

XI Procedure

1. Make connections as shown in the respective circuit diagram in Fig No.1 & Fig No.2 using breadboard
2. Connect +5 V to pin 14 and connect ground to pin no.7 of all IC's used.
3. Apply inputs as shown in observation table No.1 and observe the output on LED.
4. Note down the output in the observation table.

[Space for answer]

XII Resources Used

S. No.	Name of Resource	Broad Specifications		Quantity	Remarks (If any)
		Make	Details		
1.	IC				
2.					
3.					

XIII Actual Procedure Followed

………

………

………

XIV Precautions Followed

………

………

………

XV Observations and Calculations

Inp		Outputs	
A	B	LHS=$\overline{A + B}$	RHS= $\bar{A} . \bar{B}$
0(0V)	0(0		
0(0V)	1(5		
1(5V)	0(0		
1(5V)	1(5		

Inp		Outputs	
A	B	LHS= $\overline{A . B}$	RHS= $\bar{A} + \bar{B}$
0(0V)	0(		
0(0V)	1(		
1(5V)	0(		
1(5V)	1(		

XVI Results

………

………

………

XVII Interpretation of Results (Give meaning of the above obtained results)

………

………

………

………

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

....

....

....

XIX Practical Related Questions

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

1. Realize Ex-OR gate using NAND Gate..
2. Use De Morgan's Theorem to minimize
   - $\overline{\overline{AB} + \bar{A} + AB}$
3. Realize Ex-NOR gate using NOR Gate..
4. Realize Ex-OR gate using NAND Gate..
5. Draw 3-input NOR gate using 2-input NOR gates
6. Draw logical circuit for following Boolean expressions using basic gates. $Y = \overline{\overline{AB} + \bar{A} + AB}$

[**Space for answer**]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://www.allaboutcircuits.com/textbook/digital/chpt-7/demorgans-theorems/
2. https://www.youtube.com/watch?v=km66mgxTk78
3. https://www.youtube.com/watch?v=RrynEQ7sG5A
4. https://www.youtube.com/watch?v=fGeDAgrDAms

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

Names of Student Team Members

1. …………………………...
2. …………………………..
3. …………………………..
4. …………………………..

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

Practical No. 4: Convert given expression to sum of product (sop) form using basic logic gates.

I Practical Significance

These standard forms of Boolean functions aid the logic circuit designer by simplifying the derivation of the function to be implemented. The goal of logic expression minimization is to find an equivalent of an original logic expression that has fewer variables per term, has fewer terms and needs less logic to implement. The minimization will result in reduction of the number of gates (resulting from less number of terms) and the number of inputs per gate (resulting from less number of variables per term) the minimization will reduce cost, efficiency and power consumption.

II Relevant Program Outcomes (POs)

1. **PO 1. Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2. Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 3. Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
4. **PO 4. Engineering tools:** Apply appropriate Information Technology related techniques/tools with an understanding of the limitations
5. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III Competency and Skills

This practical is expected to develop the following skills in students:

‘Build/ test digital logic circuits consist of digital ICs.

i) Test the Digital Systems, Logic Families and logic gate.

ii) Construct combinational logical circuit

IV Relevant Course Outcome(s)

i. Test the Digital Systems, Logic Families and logic gates

ii. Construct combinational logical circuit

V Practical Outcome

Convert given expression to Sum of Product (SOP) form using basic logic gates.

VI Relevant Affective domain related Outcome(s):

- Follow Safety Practices
- Handle IC and equipment carefully

VII Minimum Theoretical Background

Refer reference 3, for Real time example for better understanding.

Sum of Products (SOP) form

Various AND terms are O Red together. Each AND term may be a single Variable or a product of multiple variables (each variable may be either in Complemented or un-complemented form).

VIII Circuit diagram

- **Demonstrating an example of three inputs where at least two out of three inputs are high then output will be 1.**

Table No.4.1: Truth table for 3 inputs

Inputs			Output
A	**B**	**C**	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Using Truth table output Equation as follows:

$$\text{Output} = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Design of logic gate circuit based on the above expression::

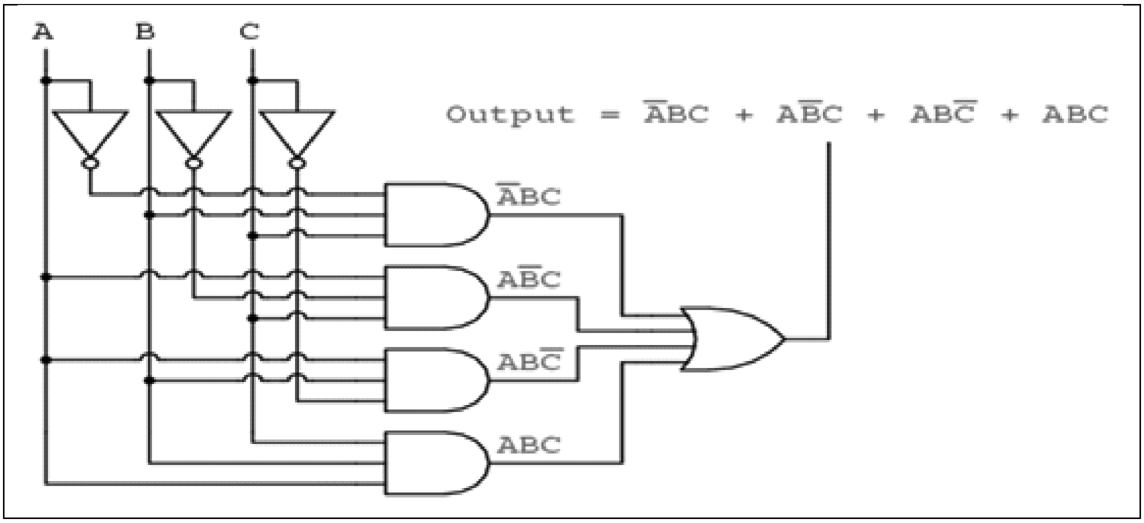

A
B
C
$\text{Output} = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$
$\overline{A}BC$
$A\overline{B}C$
$AB\overline{C}$
ABC

Figure No. 4.1 : Sum of Products (SOP) form

Figure No.4.1 circuit is quite complex, and could benefit from simplification. Using Boolean algebra techniques, the expression is significantly simplified and the circuit is as shown in figure no4.2.

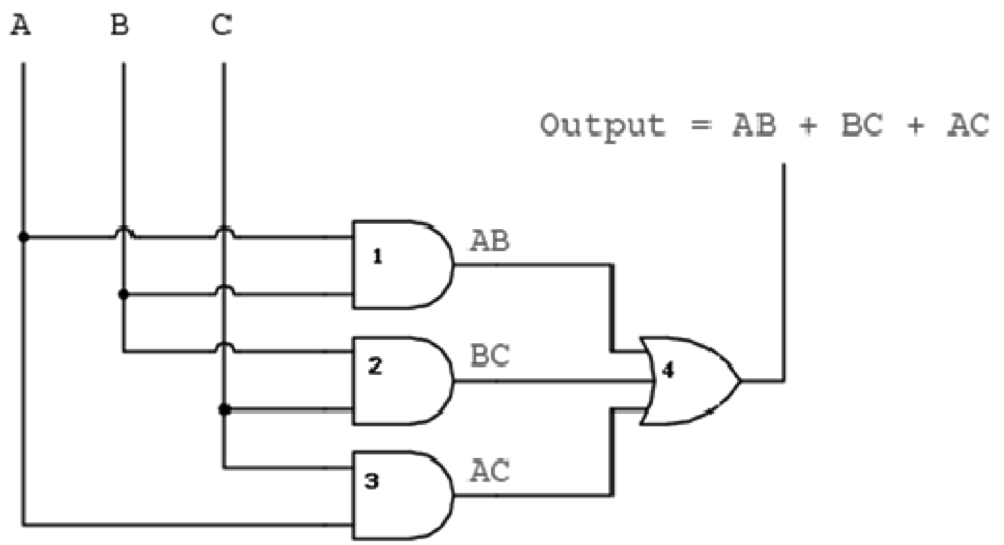


Figure No.4.2: Simplified Sum of Products (SOP) form

IX Resources Required

S. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboard	General Purpose Breadboard	1
3	DC power supply	+5 V Fixed power supply	1
4	IC	7404,7408,7432	1 Each
5	LED, Resisters	Red color 5 mm ,330R	1 Each
6	Connecting wires	Single strand 0.6 mm Teflon coating	LS
7	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC’s	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply (0-5V/500mA)

XI Procedure

1. Make connections as shown in the respective circuit diagram in **Fig No.2** using breadboard
2. Connect +5 V to pin 14 and connect ground to pin no.7 of all IC's used.
3. Apply inputs as shown in observation table No.2 and observe the output on LED.
4. Note down the output in the observation table No.2.

XII Resources Used

S. No.	Name of Resource	Broad Specifications		Quantity	Remarks (If any)
		Make	Details		
1.	IC				
2.					
3.					

XIII Actual Procedure Followed

....

....

....

XIV Precautions Followed

....

....

....

XV Observations and Calculations

Table No.4.2: Truth table

Inputs			Output			
A	B	C	Output at Gate1 (AB)	Output at Gate 2(BC)	Output at Gate 3(AC)	Output at Gate 4
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

XVI Results

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teachers <u>must design</u> more such questions so as to ensure the achievement of identified CO.

1. Define Minterm.
2. What do you mean by Canonical form?
3. Explain Significance of SOP
4. Standardize following Boolean expression.

$$Y = AC + \bar{B}C$$

[Space for answer]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://www.allaboutcircuits.com/worksheets/sum-of-products-and-product -of-sums-expressions/
2. www.youtube.com/watch?v=sp5u58Ao85U
3. https://www.allaboutcircuits.com/textbook/digital/chpt-7/converting-truth- tables-boolean-expressions/ (Strongly suggested)

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Names of Student Team Members

1. ………………………..
2. ………………………..
3. …………………………
4. …………………………

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

Practical No. 5: Convert given expression to product of sum (pos) form using basic logic gates.

I Practical Significance

These standard forms of Boolean functions aid the logic circuit designer by simplifying the derivation of the function to be implemented. The goal of logic expression minimization is to find an equivalent of an original logic expression that has fewer variables per term, has fewer terms and needs less logic to implement. The minimization will result in reduction of the number of gates (resulting from less number of terms) and the number of inputs per gate (resulting from less number of variables per term)

• The minimization will reduce cost, efficiency and power consumption.

II Relevant Program Outcomes (POs)

1. **PO 1. Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2. Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 3. Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
4. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III Competency and Skills

This practical is expected to develop the following skills in students:

‘Build/ test digital logic circuits consist of digital ICs.

1. Identify pin configuration of logic gate IC’s.
2. Utilize the functionality of the logic gates.
3. Implementation of logic circuits for given Boolean expression

IV Relevant Course Outcome(s)

i. Test the Digital Systems, Logic Families and logic gates

ii. Construct combinational logical circuit

V Practical Outcome

Convert given expression to Product of Sum (POS) form using basic logic gates.

VI Relevant Affective domain related Outcome(s):

- Follow Safety Practices
- Handle IC and equipment carefully

VII Minimum Theoretical Background

- **Refer reference 3 for Real time example for explanation of Sum of Product**

Product of Sums (POS) form

In this form, Boolean expression is defined by product of sum terms. Various OR terms are ANDed together. Each OR term may be a single variable 3-4 or a sum of multiple variables (each variable may be either in complemented or un-complemented form).

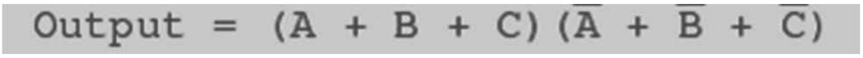
VIII Circuit diagram

- **Demonstrating an example of three inputs where at least two out of three inputs are high then output will be 1.**

Table No.5.1: Truth table for 3 inputs

Inputs			Output
A	**B**	**C**	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Using Truth table output Equation as follows:



$$\text{Output} = (A + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Design of logic gate circuit based on the above expression as follows:

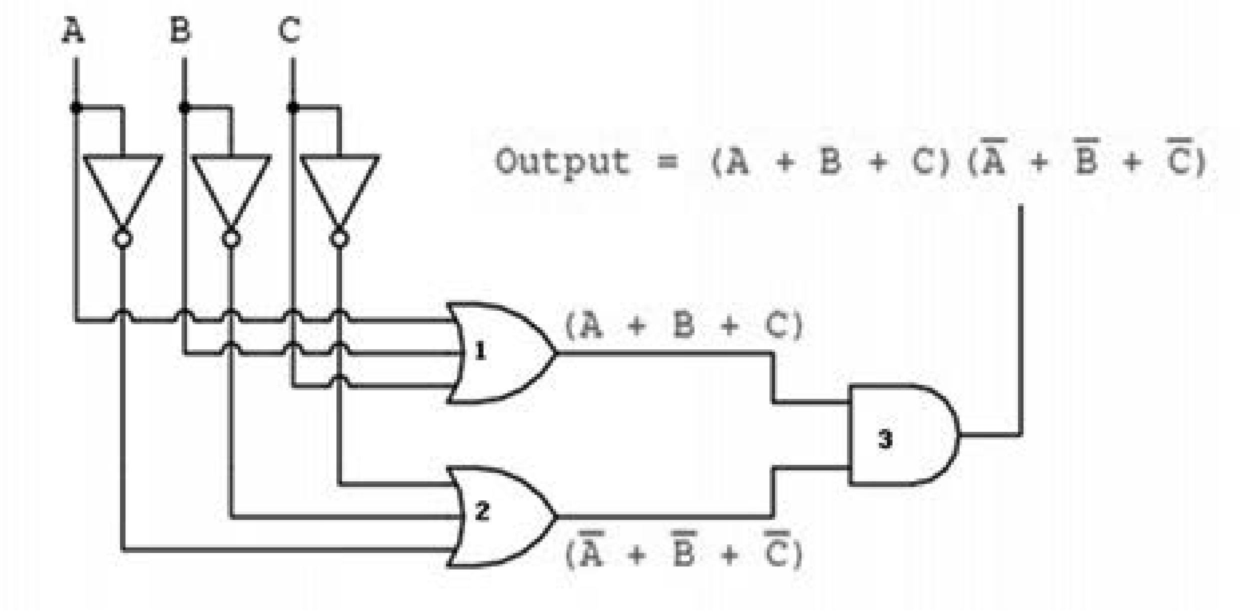

A
B
C
Output = (A + B + C)($\bar{A}$ + $\bar{B}$ + $\bar{C}$)
(A + B + C)
1
2
3
($\bar{A}$ + $\bar{B}$ + $\bar{C}$)

Figure No.5.1: Product of Sums (POS) form

IX Resources Required

S. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboard	General Purpose Breadboard	1
3	DC power supply	+5 V Fixed power supply	1
4	IC	7404,7408,7432	1 Each
5	LED, Resisters	Red color 5 mm ,330R	1 Each
6	Connecting wires	Single strand 0.6 mm Teflon coating	LS
7	Digital IC Tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC’s	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply (0-5V/500mA)

XI Procedure

1. Make connections as shown in the respective circuit diagram in Fig No.2 using breadboard
2. Connect +5 V to pin 14 and connect ground to pin no.7 of all IC's used.
3. Apply inputs as shown in observation table No.2 and observe the output on LED.
4. Note down the output in the observation table No.2.

XII Resources Used

S. No.	Name of Resource	Broad Specifications		Quantity	Remarks (If any)
		Make	Details		
1.	IC				
2.					
3.					

XIII Actual Procedure Followed

....

....

....

XIV Precautions Followed

....

....

....

XV Observations and Calculations

Table No.5.2: Truth table

Inputs			Outputs		
A	B	C	Output at Gate1	Output at Gate 2	Output at Gate 3
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

XVI Results

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

....

....

....

XIX Practical Related Questions

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

1. Define Maxterm.
2. Explain Significance of POS
3. The expression Y=(A+B)(B+C)(C+A) shows the __________________operation.
   a) AND b) POS c) SOP d) NAND
4. Standardize following Boolean expression.
   $$Y = (A + B)(\bar{A} + C)$$
5. Draw logical circuit for following Boolean expressions using basic gates.
   $$\underline{Y(A,B,C) = (A+B+C).(A+B+\bar{C}).(\bar{A}+B+C).(\bar{A}+\bar{B}+C)}$$

(Space for answers)

....

....

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://www.allaboutcircuits.com/worksheets/sum-of-products-and-product -of- sums-expressions/
2. www.youtube.com/watch?v=hSgeyLSmlSg
3. https://www.allaboutcircuits.com/textbook/digital/chpt-7/converting-truth- tables-Boolean-expressions/ (Strongly suggested**)**

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Names of Student Team Members

1. …………………………..
2. ………………………….
3. …………………………..
4. …………………………..

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

Practical No. 6: Implement Combinational Circuit using Multiplexer

I Practical Significance

To perform 1 bit Binary addition and 1 bit Binary subtraction in the digital circuit half adder and half Subtractor is used. The multiplexer tree can be used to design 2, 4, 8 bit adder and Subtractor.

II Relevant Program Outcomes (POs)

PO 1. Basic knowledge: Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.

PO 3. Experiments and practice: Plan to perform experiments, practices and to use the results to solve Information Technology related problems.

PO 4. Engineering tools: Apply appropriate Information Technology related techniques/tools with an understanding of the limitations

III Competency and Practical Skills

This practical is expected to develop the following skills in students:

'Build/ test digital logic circuits consist of digital ICs.

i. Identify pin configuration of MUX 74153 IC.

ii. Test the functionality of the combinational circuit.

IV Relevant Course Outcome(s)

i. Test the Digital Systems, Logic Families and logic gates

ii. Construct combinational logical circuit

V Practical Outcome

Verify Truth Table of half Adder & Half Subtractor using 4:1 mux 74153 IC.

VI Relevant Affective domain related Outcome(s)

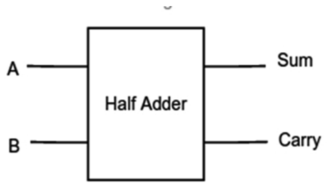
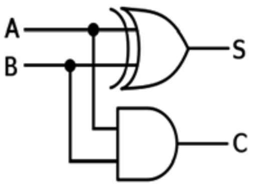
1. Handle IC and equipment carefully.
2. Follow safe practices.

VII Minimum Theoretical Background

Multiplexers can be used to implement Boolean functions of multiple variables.
Multiplexers are also known as “Data n selector, parallel to serial convertor, many to one circuit, universal logic circuit”.

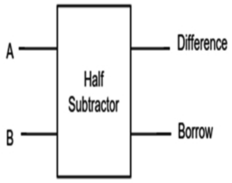
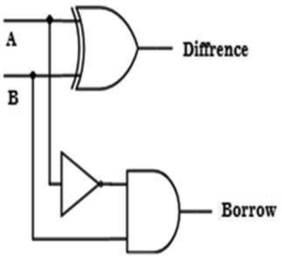
To construct any combinational circuit require the prerequisite of Sum Of product (SOP), Standard form of SOP, Minterm etc..

Table 6.1: Half Adder Implementation using logic gates

Symbol	Realization Using Gates	Truth Table
		

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0 (A0)	0 (B0)
0	1	1 (A1)	0 (B1)
1	0	1 (A2)	0 (B2)
1	1	0 (A3)	1 (B3)

Table 6.2: Half Subtractor Implementation using logic gates

Symbol	Realization Using Gates	Truth Table
		

INPUTS		OUTPUTS	
A	B	DIFFRENCE	BORROW
0	0	0 (A0)	0 (B0)
0	1	1 (A1)	1 (B1)
1	0	1 (A2)	0 (B2)
1	1	0 (A3)	0 (B3)

Types of Multiplexers

- 2 lines to 1 line (2:1) multiplexer.
- 4 lines to 1 line (4:1) multiplexer.
- 8 lines to 1 line (8:1) multiplexer.
- 16 lines to 1 line (16:1) multiplexer.

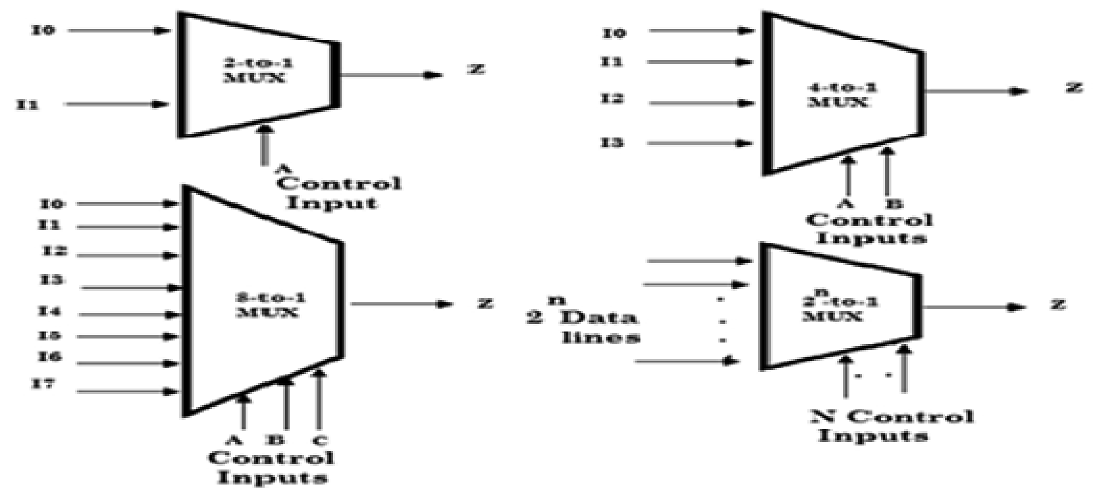
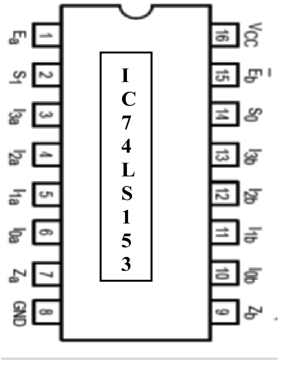


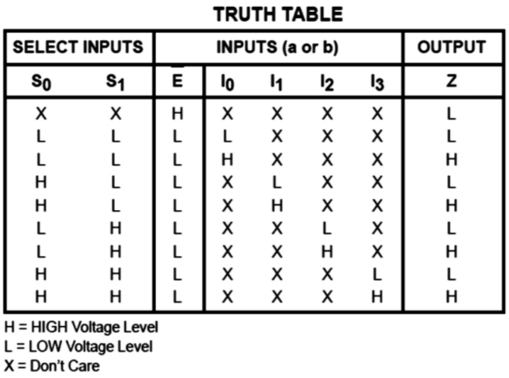
Figure 6.1 Multiplexers

Commercially available List of ICs which provide multiplexing

Table 6.3: The 74xx series has several ICs that contain multiplexer(s):		
IC No.	**IC No.**	**IC No.**
74157	Quad 2:1 mux.	Output same as input given
74158	Quad 2:1 mux.	Output is inverted input
74153	Dual 4:1 mux.	Output same as input
74352	Dual 4:1 mux.	Output is inverted input
74151A	8:1 mux.	Both outputs available (i.e., complementary outputs)
74151	8:1 mux.	Output is inverted input
74150	16:1 mux.	Output is inverted input

Table 6.4: DUAL 4-INPUT MULTIPLEXER SN54/74LS153





TRUTH TABLE

SELECT INPUTS		INPUTS (a or b)					OUTPUT
S_0	S_1	$\overline{E}$	I_0	I_1	I_2	I_3	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

Pin Function
S0 - Common Select Input
E - Enable (Active LOW) Input
I0, I1 - Multiplexer Inputs
Z - Multiplexer Output

VIII Practical set-up /Circuit diagram

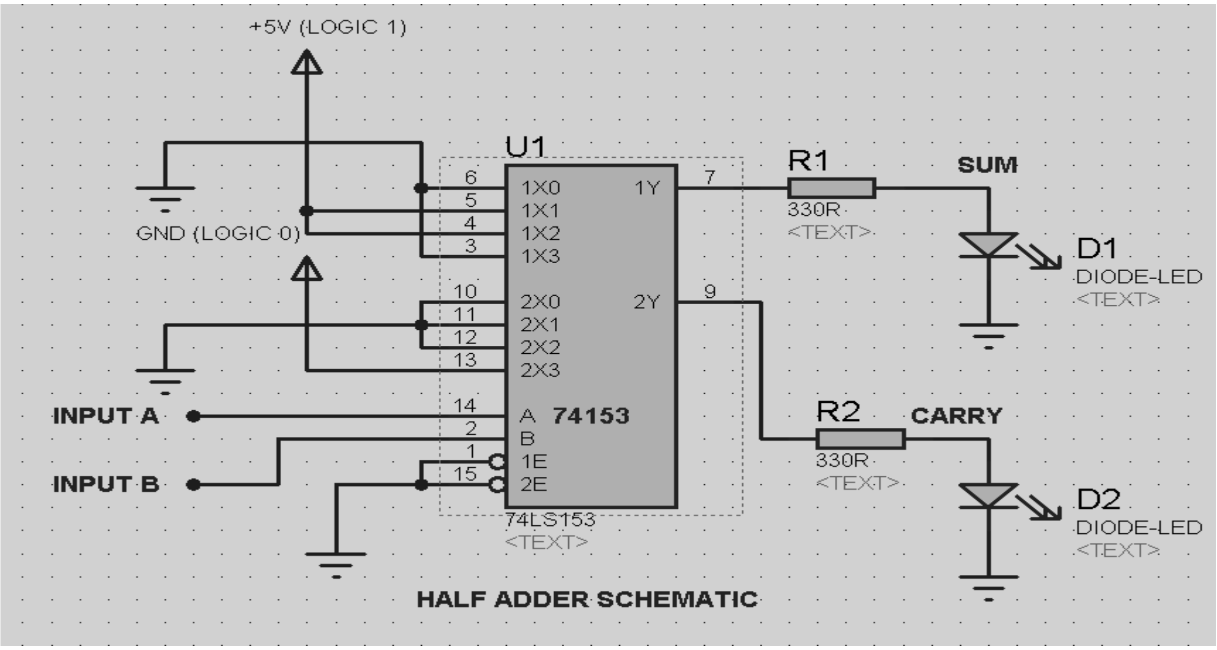


Figure 6.2: HALF ADDER Using MUX Practical set-up

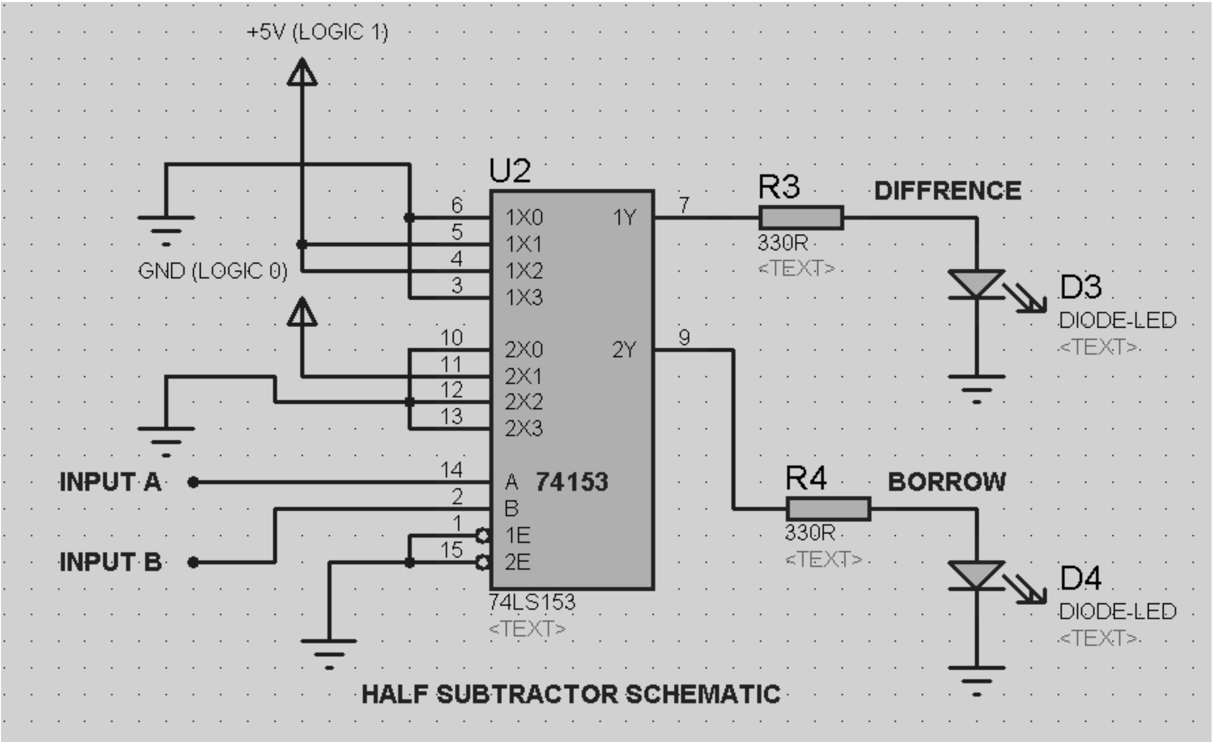


Figure 6.3: HALF SUBTRACTOR Using MUX Practical set-up

IX Resources Required

S. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboards	General Purpose Breadboards	1
3	DC power supply	+5 V Fixed power supply	1
4	IC	74153 MUX IC	1
5	LED	Red color 5 mm, Green color 5 mm	1 Each
6	Resistor	330R	2
7	Connecting wires	Single strand 0.6 mm Teflon coating	LS
8	Digital IC tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply (0-5V/500mA)

XI Procedure

1. Make the connection as per circuit diagram and give supply voltage to relevant pin
2. Connect the inputs from source to MUX IC as per logic level.
3. Observe the output on LED for each combination of input as per truth table.
4. Verify the truth table.

XII Resources Used

S. No.	Name of Resource	Broad Specifications		Qty	Remarks (If any)
		Make	Deta		
1.					
2.					
3.					
4.					

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

1. …………………………………………………………………………………...
2. …………………………………………………………………………………...
3. …………………………………………………………………………………...

XIV Precautions Followed

1. …………………………………………………………………………………...
2. …………………………………………………………………………………...

XV Observations and Calculations (use blank sheet provided if space not sufficient)

Inputs		Half Adder		Half Subtractor	
		SUM	CARRY	DIFFRENCE	BORROW
A	B	LED Status (ON/OFF)	LED Status (ON/OFF)	LED Status (ON/OFF)	LED Status (ON/OFF)
0(0V)	0(0V)				
0(0V)	1(5V)				
1(5V)	0(0V)				
1(5V)	1(5V)				

XVI Results

1. ………………………………………...
2. ………………………………………………………………………………….

XVII Interpretation of Results (Give meaning of the above obtained results)

………………………………………………………………………………………
………………………………………………………………………………………

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

………………………………………………………………………………………
………………………………………………………………………………………

XIX Practical Related Questions

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

1. The minimum number of 2:1 multiplexers are required to realize a 4:1 multiplexer is:…….
   a. 1 b. 2 c. 3 d. 4
2. Implement NOT, AND, OR gate using 2:1 MUX
3. Implement 4 : 1 MUX using 2 : 1 MUX
4. What is the function of an enable input on a multiplexer chip?
5. Give the application of multiplexer.
6. How many select lines will 32:1 MUX will have?
7. Why MUX is called as data selector.

(Space for answer)

………

………

………

………

………

………

………

………

XX References / Suggestions for further reading

1. https://www.electronicshub.org/multiplexer-and-demultiplexer
2. https://www.youtube.com/watch?v=1WaxJ9TChns
3. http://www.gatepaper.in/2014/11/previous-gate-questions-on- multiplexers.html
4. https://www.geeksforgeeks.org/multiplexers-digital-electronics/

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Names of Student Team Members

1. …………………………
2. …………………………
3. …………………………
4. …………………………

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

Practical No. 7: Construct S-R, J-K, D and T flip-flop and Verify their Truth Tables

I Practical Significance

Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers as memories, registers and counters and in many other types of systems.

II Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2. Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 3. Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
4. **PO 4. Engineering tools:** Apply appropriate Information Technology related techniques/tools with an understanding of the limitations.
5. **PO 10: Life-long learning**: Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III Competency and Practical Skills

This practical is expect to develop the following skills in students

'Build/ test digital logic circuits consist of digital ICs.'

i. Identify Flip-flop type.
ii. Make connections according to circuit.
iii. Test the functionality of the Flip-flop circuit.

IV Relevant Course Outcome(s)

i. Test the Digital Systems, Logic Families and logic gates
ii. Construct Sequential logical circuit

V Practical Outcome

Construct S-R, J-K, D and T flip-flop and verify their truth tables.

VI Relevant Affective domain related Outcome(s)

1. Handle IC and equipment carefully.
2. Follow safe practices.

VII Minimum Theoretical Background

Flip-flops and latches are used as data storage elements. FLIP-FLOPs do not change state at exactly the same time or only one FLIP-FLOP changes state for any clock pulse.

Types of Flip Flops

- SR Flip Flop
- JK Flip Flop
- D Flip Flop
- T Flip Flop

Commercially available List of ICs which provide multiplexing

The 7400 series has several ICs that contain Flip Flops

Part number	Description
7473	Dual J-K Flip-Flop, Asynchronous Clear
7474	Dual D Positive Edge Triggered Flip-Flop, Asynchronous Preset And Clear
7475	4-Bit Bi-stable Latch
7476	Dual J-K Flip-Flop, Asynchronous Preset And Clear

VIII Practical set-up Circuit diagram

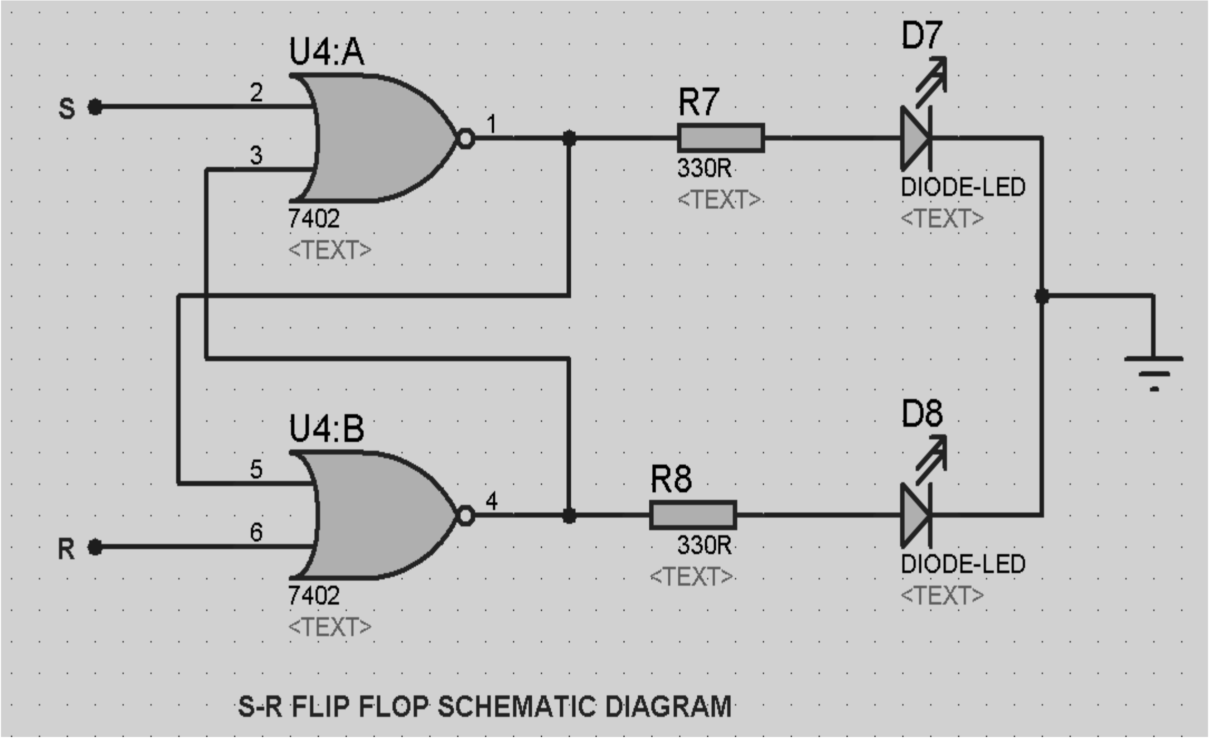


Figure 7.1 Practical set-up for S-R Flip Flop

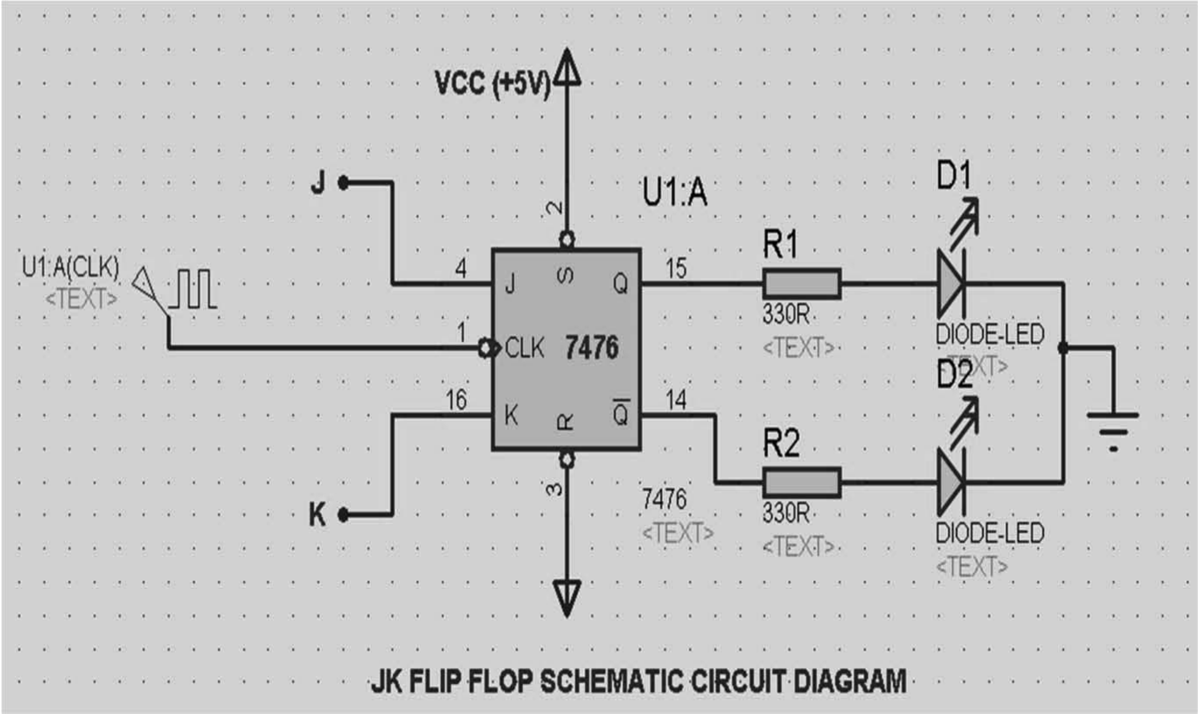


Figure 7.2 Practical set-up for J-K Flip Flop

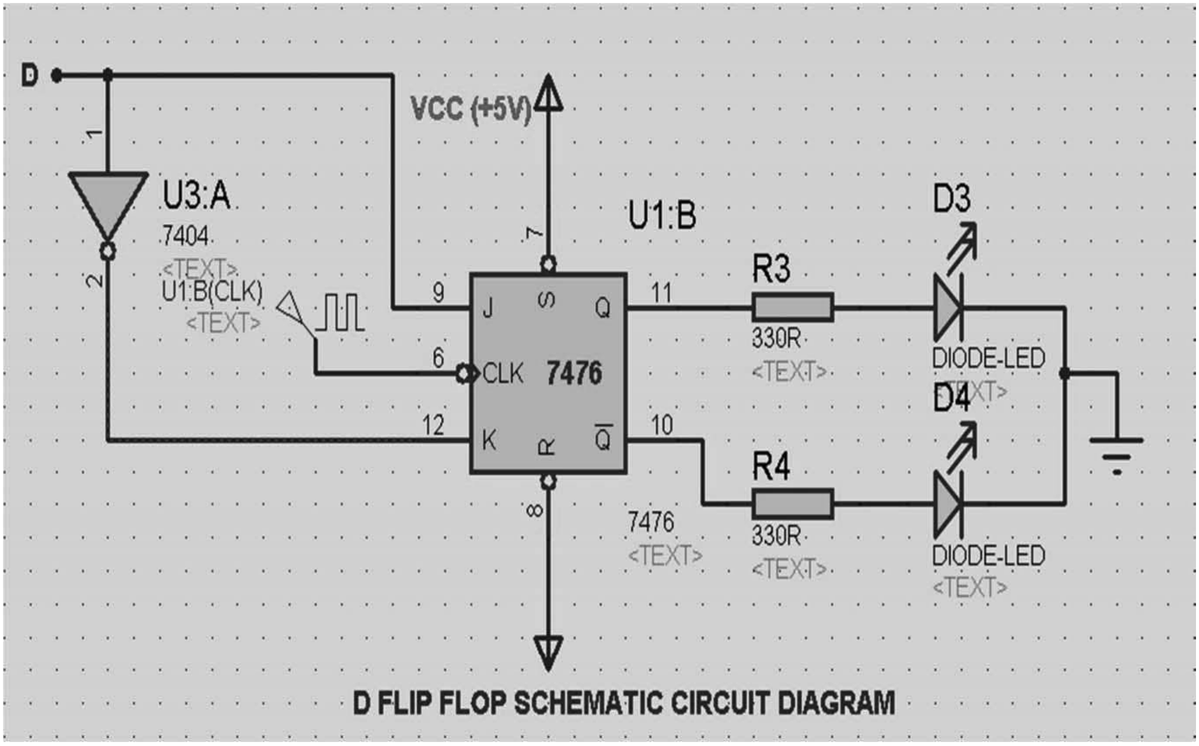


Figure 7.3 Practical set-up for D Flip Flop

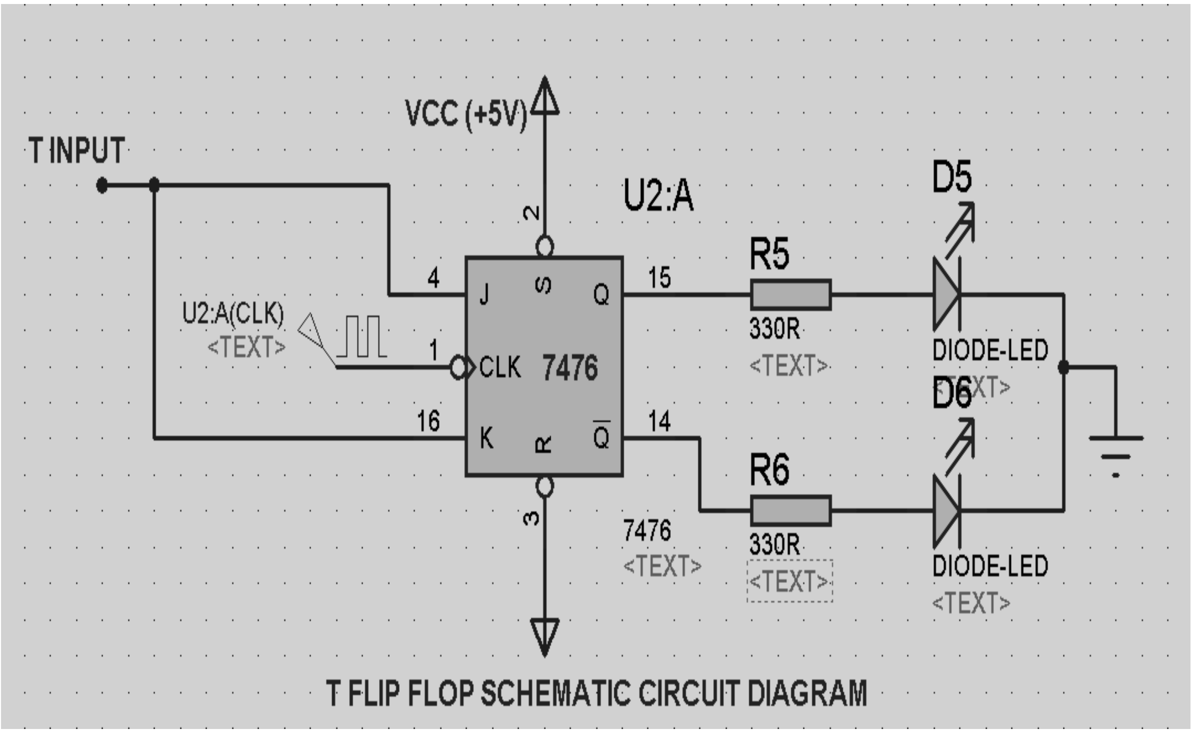


Figure 7.4 Practical set-up for T Flip Flop

IX Resources Required

S. No.	Name of Resource	Suggested Broad Specification	Quantity
1	Digital Multimeter	Digital Multimeter: 3 1/2 digit display.	2
2	Breadboards	General Purpose Breadboards	1
3	DC power supply	+5 V Fixed power supply	1
4	IC	7476,7402	1 Each
5	LED	Red color 5 mm, Green color 5 mm	1 Each
6	Connecting wires	Single strand 0.6 mm Teflon coating	LS
7	Digital IC tester	Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS Ic's	1

X Precautions to be followed

1. Test the IC using digital IC tester before conducting the experiment
2. Check Circuit connections before switch on the power supply
3. Give suitable power supply (0-5V/500mA)

XI Procedure

1. Make the connection as per circuit diagram and give supply voltage to relevant pin
2. Connect the inputs from source to IC 7476 and IC 7402 as per logic level.
3. Observe the output on LED for each combination of input as per truth table.
4. Verify the truth table.

(Space for answer)

……

……

…… \

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XII Resources Used

S. No.	Name of Resource	Broad Specifications		Quantity	Remarks (If any)
		Make	Details		
1.					
2.					
3.					
4.					

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

………

………

………

XIV Precautions Followed

………

………

………

XV Observations and Calculations (use blank sheet provided if space not sufficient)

SR F/F				JK F/F			
INPUTS		OUTPUTS		INPUTS		OUTPUTS	
S	R	Q	$\overline{Q}$	J	K	Q	$\overline{Q}$
0 (0V)	0 (0V)						
0 (0V)	1 (5V)						
1 (5V)	0 (0V)						
1 (5V)	1 (5V)						

D F/F					D F/F				
INPUTS			OUTPUTS		INPUTS			OUTPUTS	
D	J	K	Q	$\overline{Q}$	T	J	K	Q	$\overline{Q}$

XVI Results

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

XVIII Conclusions and recommendations (Actions/decisions to be taken based on the interpretation of results).

....

....

....

XIX Practical Related Questions

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

1. List four Basic Flip-flop applications**?**
2. Write the difference between a flip-flop & latch?
3. One example of the use of an S-R flip-flop is as:
   a) Transition pulse generator b) Racer
   c) Switch debouncer d) Astable oscillator
4. What is a trigger pulse?
5. A basic S-R flip-flop can be constructed by cross-coupling of which basic logic gates?
6. Write the significance of the J and K terminals on the J-K flip-flop?
7. The circuits of NOR based S-R latch classified as asynchronous sequential circuits, why?
8. How many flip-flops are in the 7476 IC?
9. Give reason of D flip-flops receives its designation or nomenclature as ‘Data Flip-flops’?
   a) Due to its capability to receive data from flip-flop
   b) Due to its capability to store data in flip-flop
   c) Due to its capability to transfer the data into flip-flop
   d) All of the Mentioned
10. Give Two applications of T flip flop

[Space for answer]

....

....

....

....

....

....

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

XX References / Suggestions for further reading

1. https://electronicsforu.com/resources/learn-electronics/flip-flop-rs-jk-t-d
2. http://www.learnerswings.com/2014/07/animated-tutorial-of-7473-dual-master.html

XXI Suggested Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

Names of Student Team Members

1. ………………………..
2. ………………………..
3. …………………………
4. …………………………

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

Practical No. 8: Evaluate Arithmetic Operations

I. Practical Significance

- Assembly language programming is useful to aware programming environment, and development of code and debugging and execution skills. It is easier to understand and save lot of time of the programmer, and code optimization.

II. Relevant Program Outcomes (POs)

1. **PO 2: Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 4: Engineering tools:** Apply appropriate Information Technology related techniques/tools with an understanding of the limitations.
4. **PO 10: Life-long learning***:* Engage in independent and life-long learning along with the technological changes in the IT and allied industry.
5. **PO 7: Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of practice in the field of Information Technology.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

- **Build digital systems including microprocessor based systems**

1. Write algorithm and draw flow Chart.
2. Assemble, link and debug assembly language program s.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.
ii) Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Write and execute an Assembly Language Program (ALP) to add / subtract two 8 bit and 16 bit numbers with the help of programming tools and any simulator.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.

VII. Minimum Theoretical Background

Assembly Language

- Assembly Language is low level programming language.
- It is used to write program statements in the form of mnemonics (English like words those specify certain action to be performed) , variables and /or operands to form instructions.
- Assemble translates assembly language program into machine language.

Table 8.1: Assembly Language Program Statements Format

Label Field	Opcode Field	Operand Field	Comment Field
NEXT:	ADD	AL,07H	;add 07H in contents of AL register

Field Position	Description
First field	Label field is a name which is mapped to an address when the program is assembled and loaded in the memory for execution (When required and specified).
Second field	Opcode field contains the mnemonic for the Instruction to be performed
Third field	Operand field contains data, memory addresses, port addresses or names of registers on which the instruction is performed
Fourth field	Comment field indicates the function of the instruction

Program Coding format: Assembly Language Program Structure

	code segment	; Start of code segment
Start:	assume cs:code, ds:data, es:extra, ss:stack	; assign names to logical segments
	mov dx, data	; initialize segment registers
	mov ds, dx	
	………	;program instructions
	………	
	………	
	mov ah,4cH	; Terminate the Program with a return code
	int 21H	
	code ends	
	data segment	
	Num1 db 10h	; assign values to different data types in data segment
	Num2 db 20h	
	data ends	
	Extra segment	**;OPTIONAL / as an when needed**
	………	; assign values to different data types in Extra segment

	…………..	
	Extra ends	
	Stack segment	**;OPTIONAL / as an when needed**
	………	; assign values to different data types in Stack segment
	…………..	
	Stack ends	
	End	; End of the program

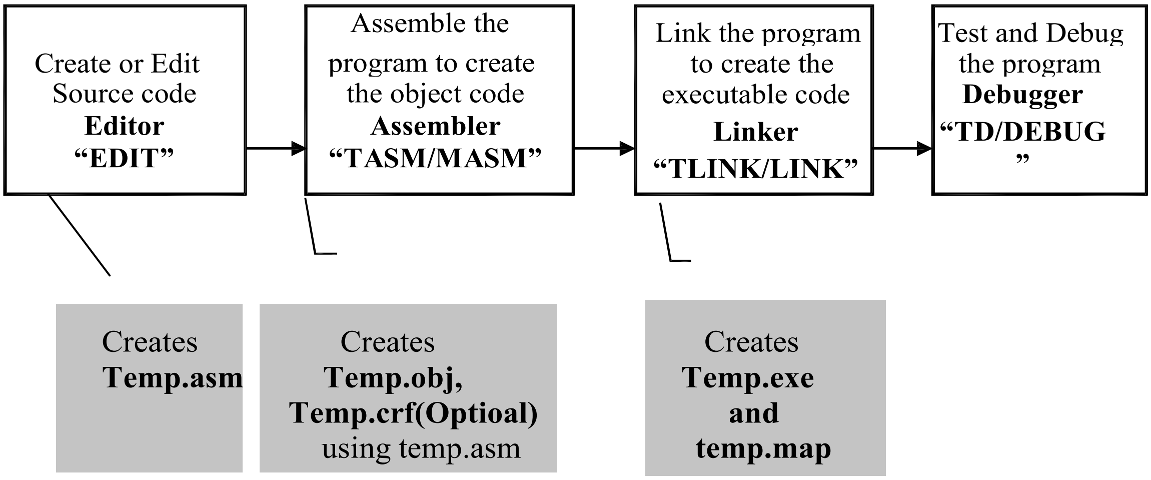

Create or Edit Source code
Editor
"EDIT"
Assemble the program to create the object code
Assembler
"TASM/MASM"
Link the program to create the executable code
Linker
"TLINK/LINK"
Test and Debug the program
Debugger
"TD/DEBUG"
Creates Temp.asm
Creates Temp.obj, Temp.crf(Optioal) using temp.asm
Creates Temp.exe and temp.map

Figure 8.1: Program development Procedure

Summary of tools used:

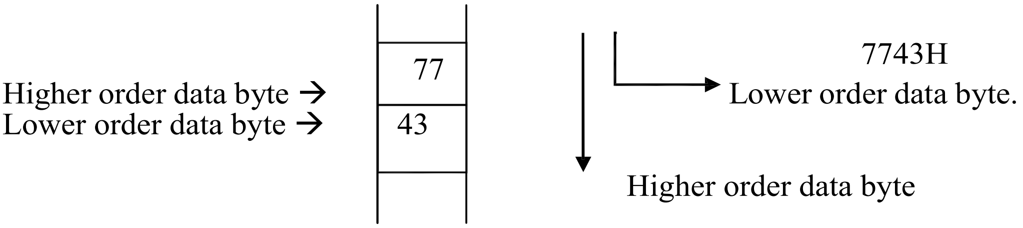
- **Editor** - EDIT FILENAME.ASM to generate source file.
- **Assembler** - TASM FILENAME.ASM generates FILENAME.OBJ , TASM FILENAME.ASM / LA generates FILENAME.LST (Optional)
- **Linker** - TLINK FILENAME.OBJ generates FILENAME.EXE and FILENAME.MAP file
- **Debugger** **-** TD FILENAME.EXE /TD FILENAME **used to test and find**
  - or Trace errors in program
  - to check contents of different registers
  - to trace transfer of control among the program

Different Number Systems

There are various number systems are supported in 8086 Assembly Language Programming: ex. Binary, Octal, Decimal, Hexadecimal, Binary Coded Decimal-BCD

Little Endean Architecture

The Intel 8086 processor follows the little Endean architecture.



Assembler Directives

- These are the directions given to the assembler or pseudo operations.
- These are used in Assembly Language programming.

Table 8.2: Summary of Assembler Directives:

Type	Size	Used for reserving/ allocating memory	Example
ASSUME	Compulsory in every program where logical segments are used	Assigns names to Logical segment(s)	ASSUME CS:CODE,DS:DATA; SS:STACK,ES:EXTRA (Registers used to access the contents of segments are CS,DS,SS,ES respectively)
SEGMENT	---	Used to indicate start of segment	CODE SEGMENT ; Start of Code Segment DATA SEGMENT ; Start of Data Segment
ENDS	---	Used to indicate the END of a segment	CODE ENDS ; End of Code Segment DATA ENDS ; End of Data Segment EXTRA ENDS ; End of Extra Segment
.DATA .CODE		used in dot programming	Simplified segment directive

Arithmetic Instructions (ADD, SUB)

ADD destination, source
(ADD register/memory, register/ memory /immediate data):

- This is an instruction used to perform addition of binary values from source to destination, and stores result in destination, AL register (in case of 8 bit no.s) and AX register (in case of 16 bit no.s)
  Destination = Source + Destination
  (Destination will be AL or AX register and source may be immediate data value or register or direct or indirect memory location where the source is stored.)
- Values may be byte, word or double word
  Flags affected are: AF, CF, OF, PF, SF and ZF
  Some examples are:

Instruction	Description
ADD AL,45H	**Immediate addressing mode** instruction that adds immediate number with AL and stores result in AL(AL =AL +45).
ADD AX,BX	**Register addressing mode** instruction that adds the contents of BX with AX and stores result in AX.

ADC destination, source:
(ADD register/memory, register/ memory /immediate data):

- This is an instruction used to perform **multi-word** addition of binary values
  ADC adds the contents of the carry Flag (CF 0/1) to operand1 , and then adds operand2 to operand1 , like ADD Instruction.
  Destination = Source + Destination + CF
  (Destination will be AL or AX register and source may be immediate data value or register or direct or indirect memory location where the source is stored.)
- Values may be byte, word or double word
  Flags affected are: AF, CF, OF, PF, SF and ZF

Some examples are:

Instruction	Description
ADC AX,4500H	**Immediate addressing mode** instruction that adds immediate number 4500H to AX with carry and stores result in AX (AX =Ax +4500 + CF (0/1)).
ADC AX,BX	**Register addressing mode** instruction that adds the contents of BX with AX with carry flag (CF) and stores result in AX.

SUB destination, source:
(SUB register/memory, register/ memory /immediate data)

- This is an instruction used to perform subtraction of binary values from source to destination, and stores result in destination, AL register (in case of 8 bit no.s) and AX register (in case of 16 bit no.s)
- Values may be byte, word or double word

Destination = Destination - Source
(Destination will be AL or AX register and source may be immediate data value or register or direct or indirect memory location where the source is stored.)
Flags affected are: AF, CF, OF, PF, SF and ZF

Some examples are:

Instruction	Description
SUB AL,75H	**Immediate addressing mode** instruction that subtracts immediate number from AL and stores result in AL(AL =AL -75).
SUB AX,BX	**Register addressing mode** instruction that subtracts the contents of BX from AX and stores result in AX.

SBB destination, source:
(SBB register/memory, register/ memory /immediate data)

- This is an instruction used to perform subtraction to carry an overflowed 1 bit into next stage of arithmetic, SBB first subtracts contents of CF (0/1) from operand1 and then subtracts operand2 from operand1 like SUB and stores result in destination, AL register (in case of 8 bit no.s) and AX register (in case of 16 bit no.s)
- Values may be byte, word or double word

Destination = Destination - Source -CF
(Destination will be AL or AX register and source may be immediate data value or register or direct or indirect memory location where the source is stored.)
Flags affected are : AF, CF, OF, PF, SF and ZF

Some examples are:

Instruction	Description
SBB AX,2334H	**Immediate addressing mode** instruction that subtracts immediate number 2324H and borrow from AX and stores result in AX (AX =Ax - 2324 - CF (0/1)).
SBB AX,BX	**Register addressing mode** instruction that subtracts the contents of BX and borrow from AX and stores result in AX.

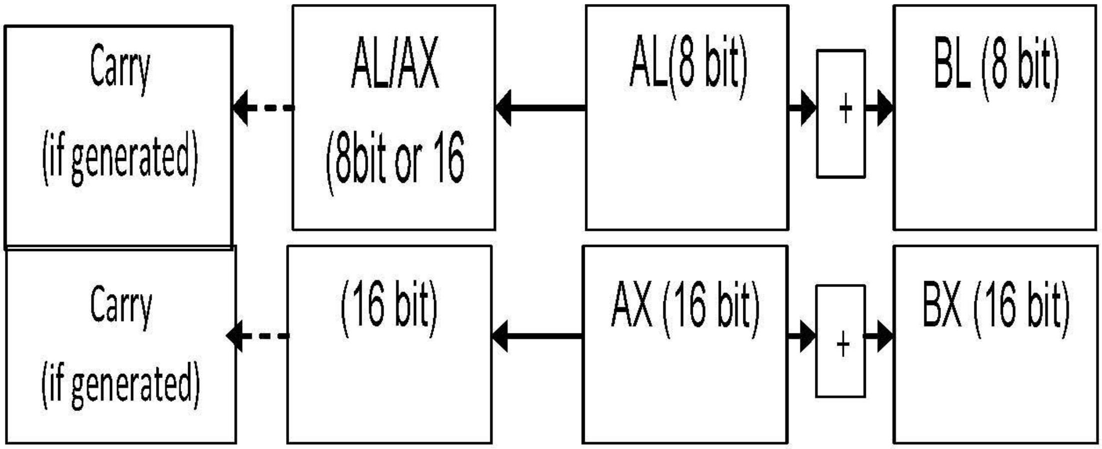


Figure 8.2: Concept diagram for Addition of 8 or 16 bit no.s

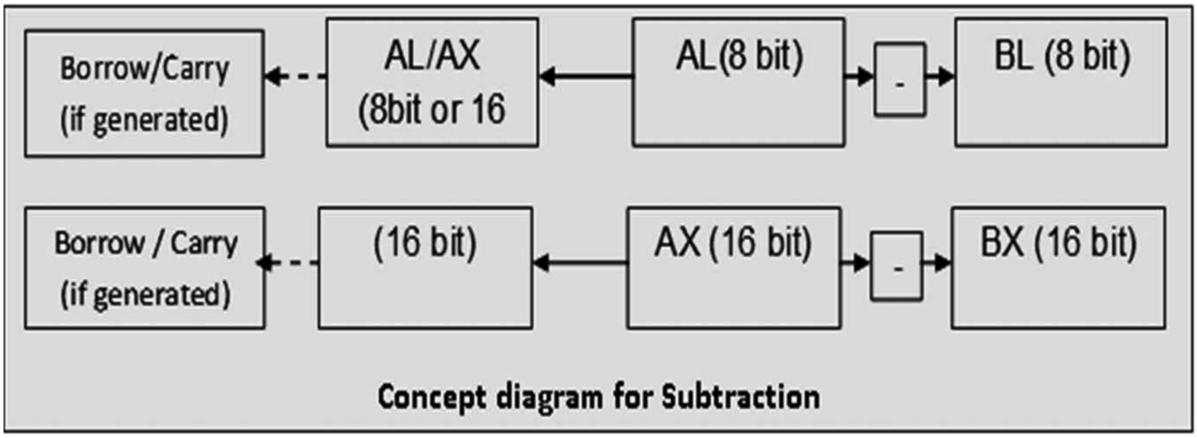


Figure 8.3: Concept diagram for Subtraction of 8 or 16 bit no.s

VIII. Algorithm

1. Initialize the Data and Code Segment.
2. Read the NUMBER1 – first 16 bits to be added.
3. Read the NUMBER2– second 16 number to be added.
4. Add the two.
5. Store the 16 bit SUM.

IX Flow Chart

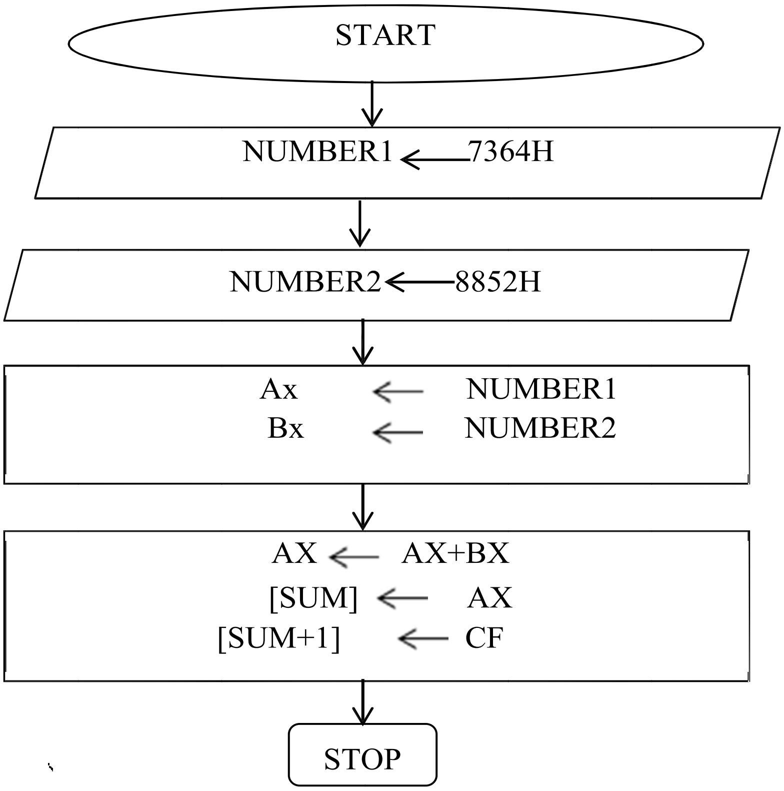


Figure 8.4: Flowchart for addition of two 16 bit numbers

X. ‘Assembly Language Program Code

Program for addition of two 16 bit numbers
DATA SEGMENT
NUM1 DW 1234H
NUM2 DW 9876H
SUM DW 2 DUP(0)
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START: MOV DX, DATA
MOV DS, DX
MOV AX, NUM1
MOV BX, NUM2
ADD AX, BX
MOV SUM, AX
JNC EXIT
MOV SUM+2, 01
EXIT: MOV AH, 4CH
INT 21H
CODE ENDS
END START

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows 7/LINUX version 5.0 or later		
3	Software	Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug (and any other suitable)		
4	LCD Projector	The arrangement of LCD Projector shall be made for quality and Effective execution of experiment.	**Suggested**	For demonstration of experiment no. 8 to 16 Assembling, linking debugging using tools / simulator

XII Precautions

1. Do not give name of assembler directives as operand names.
2. Do not give program name as any instruction name like ADD.ASM (as ADD is instruction)
3. Program name cannot be more that 8 characters and does not carry numeric as first character and black space in the name (i.e. ADD 16.ASM)
4. Use proper registers with their data size for relevant data types.

XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software / tools used	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

………
………
………

XVI Practical Related Questions

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

1. ADC instruction performs ………………………………………
2. SBB instruction performs ……………………………….. (/Subtraction with borrow/Subtraction/subtraction without carry/).
3. Contents of Zero flag affected after execution of ………………… (ADD /SUB /ADC/ All given) instruction.
4. The addressing mode of SUB AX, BX instruction is ……………….…………...
5. Assume suitable numbers and perform the following operations:
6. Using above program fill the table :

Operation	**Addition (Sample)**	**Addition with carry (Sample)**	**Addition (16bit)**	**Subtraction (16 bit)**	**Addition with carry (16 bit)**	**Subtraction with carry (16 bit)**
Destination	10H	34H				
Source	20H	27H				
Carry	--	01H				
Destination	30H	62H				

(Space for answer)

………
………
………
………
………
………
………
………
………
………
………
………
………
………

XVII Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write assembly language program for performing addition of three numbers using different variables like NUM1, NUM2, and NUM3 etc.
2. Complete the following table with the given instructions:

Sr. No.	Arithmetic Operation	Instruction Format	Example
1.	Addition (8 bit)		
2.	Addition with Carry (8 bit)	ADC Destination, Source	ADC AL,BL
3.	Subtraction (8 bit)		
4.	Subtraction with carry (8 bit)		
5.	Addition(16 bit)		
6.	Addition with Carry (16 bit)	ADC Destination, Source	ADD AX,CX
7.	Subtraction(16 bit)		
8.	Subtraction with carry (16 bit)		

3. Write the meaning of following directives:

Assembler directives	Meaning
DB	
DW	
ASSUME	
ENDS	
END	
SEGMENT	

(Space for answers)

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. https://www.tutorialspoint.com/assembly_programming (As on 19th Jan 2018)
2. https://www.elprocus.com/8086-assembly-language-programs-explanation/ (As on 19th Jan 2018)
3. http://www.pcpolytechnic.com/computer/learning.html (As on 19th Jan 2018)

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

List of Students /Team Members

1. …………………………………..

2. …………………………………..

3. …………………………………..

4. ………………………………….

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

Practical No. 9: Perform Sum of Series of Numbers

I. Practical Significance

Array addition is useful in solving many application level programming, using Assembly Language programming process, Algorithm, Flow Chart and various instructions.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 4: Engineering tools:** Apply appropriate Information Technology related techniques/tools with an understanding of the limitations.
4. **PO 7: Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of practice in the field of Information Technology.

. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

- **Build digital systems including microprocessor based systems**
  1. Write algorithm and draw flow Chart.
  2. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.
ii) Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Write and execute an ALP to find sum of series of 8 bit and 16 bit numbers.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.

VII. Minimum Theoretical Background

Instructions

- **MOV REG, REG:** Instruction to performs data transfer between specified registers or memory locations.
- **INC REG** Instruction, used to Increment value of register or memory location by 1 (REG +1)
- **DEC REG** Instruction, used decrements value of register or memory location by 1 (REG-1)
- **JNC Label:** It is conditional jump instruction used to jump to the specified location when condition satisfied, There are different such conditional instructions like JC, JZ, JNZ, JB, JAE, JANE etc. There is unconditional jump instruction used for unconditional Jump.
- **LOOP LABEL (Address):** Controls the execution of a routine a specified number of times/ count . Count is given in CX or CL.
  Loop appears at the end of LOOP and decrements CX/CL by 1.

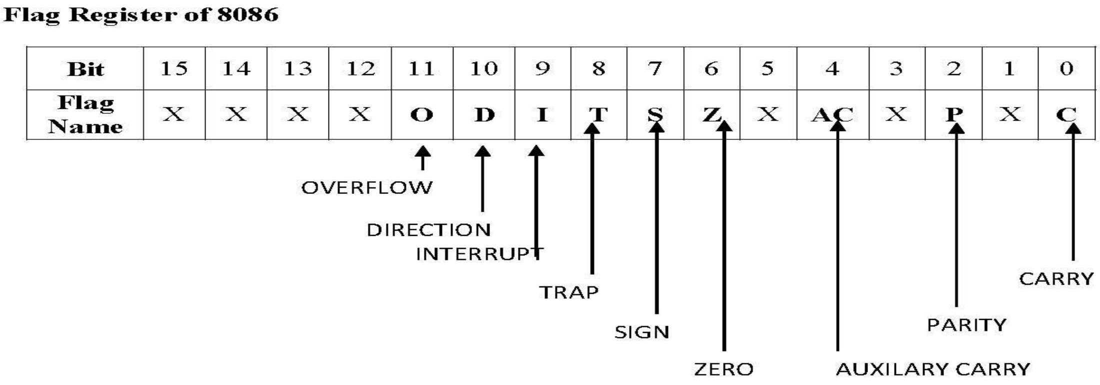
If CX is nonzero, LOOP transfers to its operand address (short Jump) otherwise LOOP drops through to the next instruction.

Array of numbers

- A Set of similar data types.
- Stored in Successive locations (Addresses).
- Easy to assess locations using array.

Array Elements	10	20	45	..	..	..	..	..	
Address of element	Array[0]	Array[1]	Array[2]	Array[3]	Array[.]	Array[..]	Array[n-2]	Array[n-1]	Array[n]

Figure 9.1: illustration of elements of array stored in consecutive memory locations



Flag Register of 8086

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flag Name	X	X	X	X	O	D	I	T	S	Z	X	AC	X	P	X	C

Figure 9.2: Position of Flag in Flag register

Table: 9.1: Description of 8086 Flags

Bit	Flag Name	Operation	If Flag = 0	If Flag = 1
0	Carry	Indicate Carry/Borrow	No Carry	Carry Generated
2	Parity	Indicate Parity of result	Operation contain even no. of 1	Operation contain Odd no. of 1
4	Auxiliary	Performs binary to BCD conversion	----	Carry is generated from lower nibble to upper nibble
6	Zero	Indicates Arithmetic or Logical operation answer as zero	If result is non-zero	If Result is Zero
7	Sign	Indicates sign magnitude of number	MSB / Result of operation is positive	MSB / Result of operation is negative
8	Trap	It is used for Single step Control (Executes one instruction at a time for debugging)	----	Program runs in single step mode
9	Interrupt	It enables or Disables Interrupt (Set by STI instruction , Clears by CLI instruction)	Maskable interrupt disabled	Maskable interrupt enabled
10	Direction	Used in String Operation. This can be set by STD and Cleared by CLD instruction.	Bytes are accessed from lower byte of memory to higher address	Bytes are accessed from higher byte of memory to lower address
11	Overflow	In case of signed arithmetic operation, not effective in unsigned operations	Result will fit in the no. bits available to accommodate	Result is too large to fit in the no. bits available to accommodate
1, 3, 5, 12-15	**Unused**	**Reserved by Intel**		

ARRAY
1000
2000
1500
1200
3500

LOOP-1	LOOP-2	LOOP-3	LOOP-4	
1000+2000=3000	3000+1500=4500	4500+1200=6700	6700+3500=9200	No Addition
Counter=04	Counter=03	Counter=02	Counter=01	Counter=00

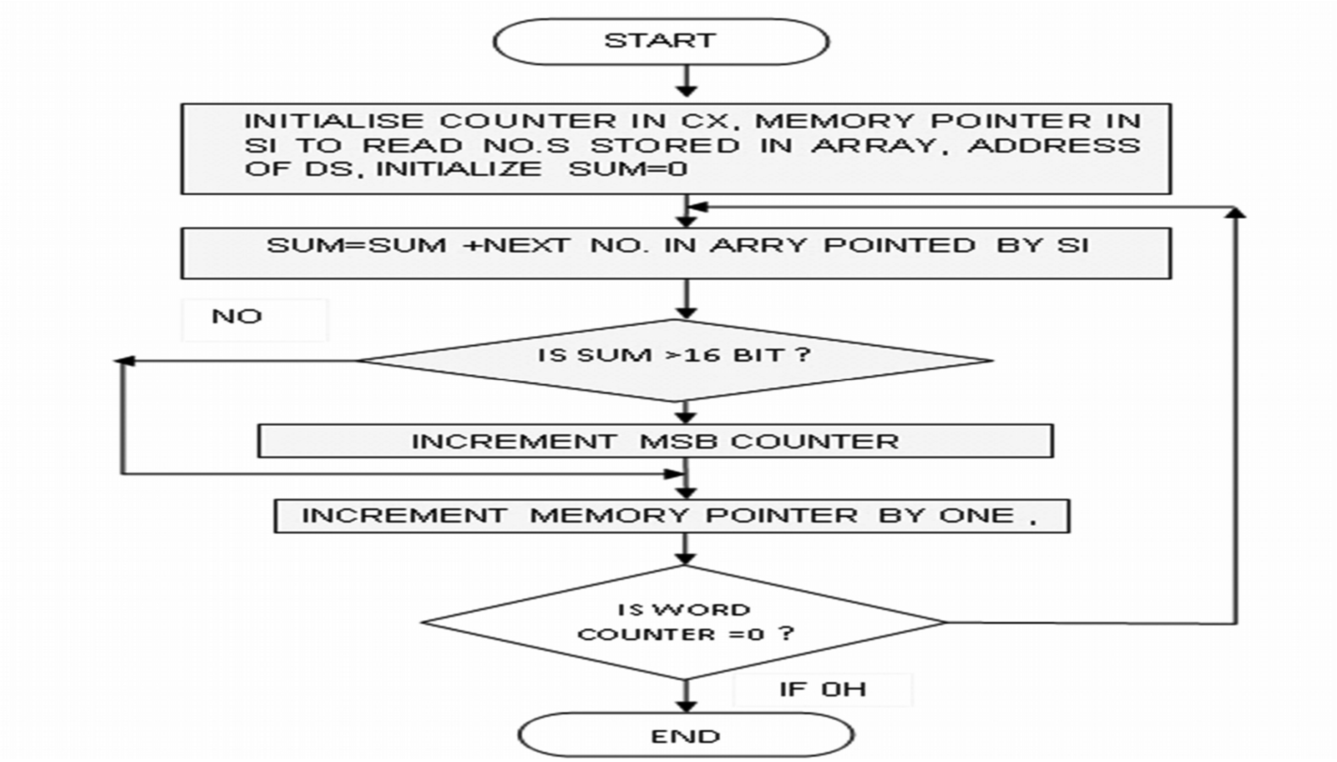
Figure 9.3: Iterations of each step of array addition

VIII Algorithm

- **Algorithm for Addition of Series of n number (16-bit)**

1. Initialize the data segment with numbers in an array to add.
2. Initialize SI (Source Index) register as a start of array which helps to access elements of array.
3. Initialize CX register as a count with N-1 and AX=0000H where N = no. of elements in an array.
4. Perform ADD instruction (AX = AX+[SI])
   (As SI =0000H it reads first element of array and AX as zero and stores addition in AX)
5. If addition is greater, then increment MSB pointer as a carry.
6. Increment SI to access next element of array.
7. Decrement the counter CX by 1.
8. Check whether the count =0 , if not go to step 6.
9. Store the result in some location or the variable declared for storing result.
10. Stop.

IX Flow Chart


START
INITIALISE COUNTER IN CX, MEMORY POINTER IN SI TO READ NO.S STORED IN ARRAY, ADDRESS OF DS, INITIALIZE SUM=0
SUM=SUM +NEXT NO. IN ARRY POINTED BY SI
NO
IS SUM >16 BIT ?
INCREMENT MSB COUNTER
INCREMENT MEMORY POINTER BY ONE .
IS WORD COUNTER =0 ?
IF 0H
END

X Assembly Language Program Code (Student shall write the program code)

XI Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII Precautions

1. Comment statements must prefix with a semicolon (;).
2. Do not give name of assembler directives as operand names.
3. Do not give program name as any instruction name like ADD.ASM (as ADD is instruction)
4. Program name cannot be more that 8 characters and does not carry numeric as first character and black space in the name (i.e. ADD 16.ASM)
5. Use proper registers with their data size for relevant data types.
6. Value of count variable shall match with data available.

XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

....

....

....

....

XVI Practical Related Questions

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

1. Effective address of Array, Num1 is available in………………
2. JNC instruction executed when Carry Flag is set / Reset.
3. Write name of flags checked while execution of JNC and JNZ instructions?
4. Complete the following table:
   Assume numbers in array which produces carry 1 and stores carry as 1 in sum upper.

	Step 1	**Register**			
		Array elements	**Sum_lower**	**Count**	**Sum_upper**
Initial situation			0000H	5	
	1			4	
	2			3	
	3			2	
	4			1	
	5			0	
					1

5. Write arithmetic instructions used in above program? List them with their syntax.
6. Explain conditional jump instructions used in program.

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

XVII Exercise

Attempt Q1. and teacher shall allot any two out of Q………and Q……… the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Here count = n-1 where n = 10.
   Enter any ten 8 bit no.s for performing addition.

ARRAY
10
20

Loop	For example	1	2	3	4	5	6	7	8	9	10
calculation	**10 +20 =30**										
Status of count		9	8	7							

2. Apply these numbers in program given above and record the corresponding results

Sr. No.	ARRAY	Case 1 (8 bit nos.)	Case 2 (16 bit nos.)
1	10		
2	20		
3	30		
4	40		
5	50		
6	60		
7	70		
8	80		
9	90		
10	100		
Total			

3. Give value of counter required to initialize to add 20 numbers?
4. Write algorithm of program to write program to multiply 5 by 4 using ADD instruction?
5. Write comments/description for instructions in program for Addition of array of five 8 bit hex numbers:

Label	Mnemonics	Descriptions/comments
	CODE SEGMENT	
	ASSUME CS: CODE, DS:DATA	
	MOV DX, DATA	
	MOV DS, DX	
	MOV CL, 05H	
	MOV AX,0000H	
	MOV SI,OFFSET ARRAY	
UP:	MOV AL, [SI]	
	ADD SUM_LOWER,AL	
	JNC NEXT	
	INC SUM_UPPER	
NEXT:	INC SI	
	LOOP UP	
	MOV AX, 4C00H	
	INT 21H	
	CODE ENDS	
	DATA SEGMENT	
	NUM1 DB 11H, 12H, 13H, 14H, 15H	
	SUM_LOWER DB 00H	
	SUM_UPPER DB 00H	
	DATA ENDS	
	END	

(Space for answers)

....

....

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. https://www.tutorialspoint.com/assembly_programming (As on 19th Jan 2018)
2. https://www.elprocus.com/8086-assembly-language-programs-explanation/ (As on 19th Jan 2018)
3. http://www.pcpolytechnic.com/computer/learning.html (As on 19th Jan 2018)
4. ANDROID application "Tutorial for assembly" (Freeware on Play store) (as on 18th Jan 2018)
5. ANDROID application "8086 Programming Tutorial" (Freeware on Play store) (as on 18th Jan 2018)
6. Video Lecture on youtube: https://www.youtube.com/watch?v=EIXtHpEumq4
7. http://nec.edu.np/faculty/chandrat/8086imp.pdf (as on 18th Jan 2018)

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
Total (25 Marks)		**100 %**

List of Students /Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..
4. …………………………………..

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

Practical No. 10: Use of assembly language in signed and unsigned Multiplication

I. Practical Significance

Multiplication is developed for equal groups situations in advance computation. Various real world problems can be solved with multiplication.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic Knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 4: Engineering tools:** Apply appropriate Information Technology related techniques/ tools with an understanding of the limitations.
4. **PO 8: Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
5. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

- **Build digital systems including microprocessor based systems**

1. Develop algorithm for implementing signed and unsigned multiplication.
2. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.
ii) Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Develop an ALP to multiply two 8 bit and 16 bit unsigned/ signed numbers.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.
3. Follow practices related to environmental context.

VII Minimum Theoretical Background

Signed Hexadecimal Numbers:

Signed Hexadecimal Numbers are represented in Sign Magnitude Form.

- Most significant bit represents the sign of the number.
  - 0 indicates a positive number
  - 1 indicates a negative number

MUL SOURCE $_{8/16}$

This instruction is used to multiply an unsigned byte (8 bits) by a byte (8 bits) or to multiply two unsigned words (16 bits).

a) Byte Multiplication:

- The 8 bit Multiplicand should be in the AL register.
- The 8 bit Multiplier should be loaded in an 8 bit register or a memory location.
- After multiplication,
  AX ← Product (16 bits).

Example:

```
MOV AL, 09H
MOV BL, 02H
MUL BL
AX=12H
```

b) Word Multiplication:

- A 16 bit multiplicand must be loaded in the AX register.
- The 16 Multiplier must be loaded in a 16 bit register or a memory location.
  After multiplication

  DX ← 16 bit Most Significant Word of the product
  AX ← 16 bit Least Significant Word of the product.

Example

```
MOV AX, 0766H
MOV BX, 4322H
MUL BX,
DX ←01F0H
AX ←AD8CH
```

IMUL SOURCE$_8$

- **IMUL SOURCE$_8$** instruction is used to multiply 8 bit signed number in source register or memory location to an 8 bit signed number in AL register. Similar to MUL instruction, the 16 bit signed result is available in AX register.

IMUL SOURCE$_{16}$

- **IMUL SOURCE$_{16}$** bit instruction is used to multiply 16 bit signed number in AX register to 16 bit signed number in source register or memory location. The 32 bit product signed result is available in DX and AX register.

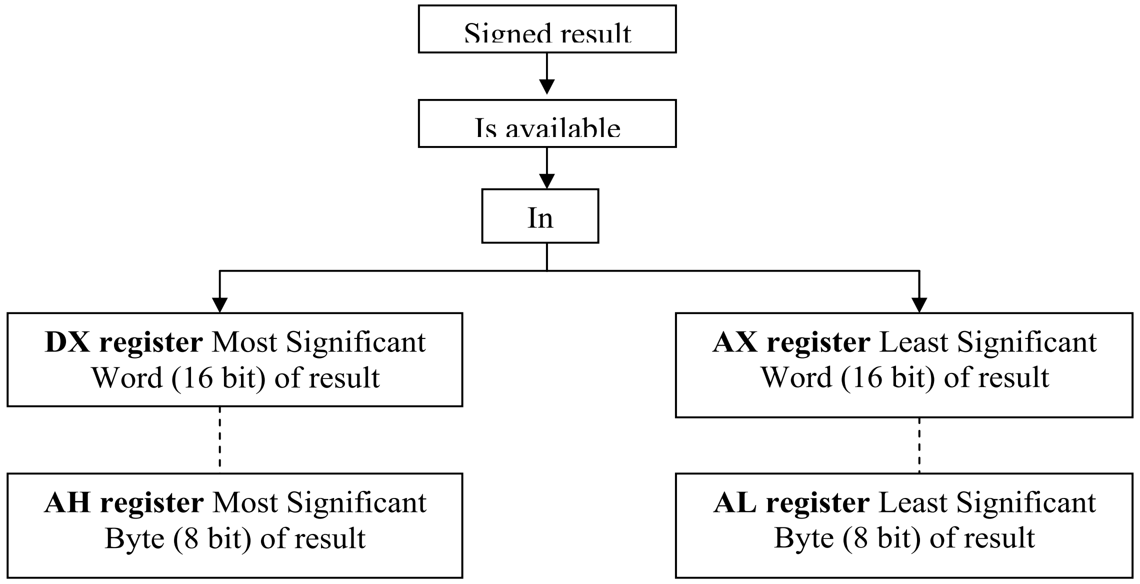

Signed result
Is available
In
DX register Most Significant Word (16 bit) of result
AX register Least Significant Word (16 bit) of result
AH register Most Significant Byte (8 bit) of result
AL register Least Significant Byte (8 bit) of result

Figure: 10.1 Concept diagram for 8 Bit By 8 Bit Multiplication:

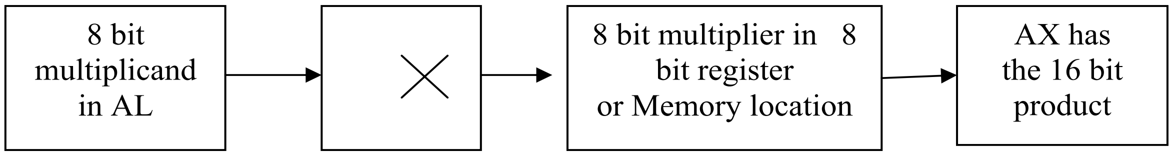

8 bit multiplicand in AL
8 bit multiplier in 8 bit register or Memory location
AX has the 16 bit product

Figure: 10.2 Concept diagram for 16 Bit By 16 Bit Multiplication:

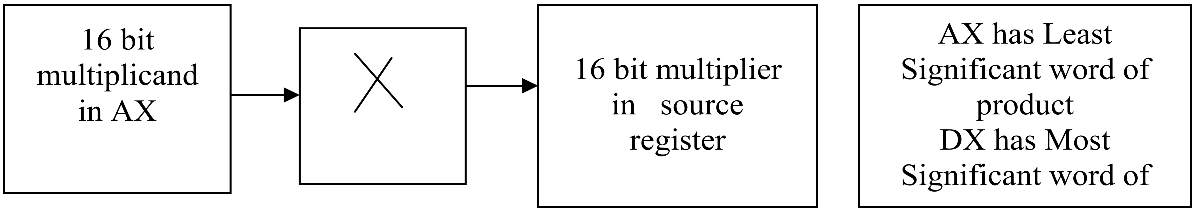

16 bit multiplicand in AX
16 bit multiplier in source register
AX has Least Significant word of product
DX has Most Significant word of

Figure 10.3 Similar structure can be extended for IMUL instruction.

VIII. Algorithm

Write the Algorithm to multiply two 8 bit no.s

Algorithm

1. Initialize the Data and Code Segment.
2. Read the 8 bit multiplicand.
3. Read the 8 bit multiplier.
4. Multiply the twon8 bit no.s
5. Store the product.
6. End.

IX Flow Chart (Student shall draw the flow chart)

X Assembly Language Program Code

Program for 16 bit by 16 bit unsigned multiplication.

```
DATA SEGMENT
NUM1  DW  2020H
 NUM2  DW  1755H
 PRO_LOW DW  ?
 PRO_HIGH DW  ?
  DATA ENDS

CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START:  MOV DX,DATA
        MOV DS,DX
        MOV AX,NUM1
        MOV BX,NUM2
        MUL BX
        MOV PRO_LOW,AX
        MOV PRO_HIGH,DX
        MOV AH,4CH
        INT 21H
```

XI Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII Precautions

1. Collect the result of multiplication in appropriate variable while transferring result to respective variable / operand.
2. Signed result of multiplication shall be verified properly.
3. Keep update of flags after multiplication.

XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. Complete the given tables:

For 8-bit multiplication

	CASE I		CASE II		CASE III	
NUMBER1	47H					
NUMBER2	53H					
MULTIPLICATION RESULT	AH =	AL =	AH =	AL =	AH =	AL =

For 16 bit multiplication

	CASE I		CASE II	
NUMBER1	4463H			
NUMBER2	7782H			
MULTIPLICATION RESULT	DX =	AX =	DX =	AX =

2. Elaborate result of multiplication for 8 bit and 16 bit multiplication in terms of AH,AL, DX,AX respectively with proper example..
3. Which instruction is used for signed multiplication?
4. What is the addressing mode of IMUL BL instruction?
5. Give this data set for your program as F4H*22H and write result ……………………………
6. For unsigned multiplication ………………………… (MUL / IMUL) instruction is used.
7. Record result 34E8H * 34H =…………………

(Space for answer)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVII Exercise

Attempt Q1. and teacher shall allot any two out of Q………and Q……… the following:

1. Complete following tables (Refer programs) .

	CASE I (For MUL)	CASE II (for IMUL)	CASE III (MUL 16 bit)
NUMBER1	40H		
NUMBER2	30H		
PRO_LOW	00H		
PRO_HIGH	12H		

2. Draw a flowchart for 8 bit by 8 bit multiplication.
3. Write assembly language program to find cube of given number.
4. Generate “Table of square for first 7 values of hexa-decimal number.
5. Student shall write program for 8 bit by 8 bit multiplication.

(Space for answers)

....

....

....

....

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. https://www.tutorialspoint.com/assembly_programming (As on 19th Jan 2018)
2. https://www.elprocus.com/8086-assembly-language-programs-explanation/ (As on 19th Jan 2018)
3. http://www.pcpolytechnic.com/computer/learning.html (As on 19th Jan 2018)
4. https://www.csie.ntu.edu.tw/~acpang/course/asm_2004/slides/chapt_07_PartII Solve.pdf (As on 19th Jan 2018)

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

List of Students /Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..
4. …………………………………..

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

Practical No. 11: Use of assembly language in signed and unsigned division

I. Practical Significance

Assembly language programming helps in simplifying complex computations, enhance speed of execution and improve logical thinking capabilities.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 4: Engineering tools:** Apply appropriate Information Technology related techniques/tools with an understanding of the limitations.
4. **PO 8:** Individual and team work
5. **PO10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

- **Build digital systems including microprocessor based systems**

1. Develop algorithm for implementing signed and unsigned Division.
2. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.
ii) Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Develop an ALP to divide two 8 bit and 16 bit unsigned/ signed numbers.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.
3. Follow practices related to environmental context.

VII. Minimum Theoretical Background

Signed Hexadecimal Numbers:

Signed Hexadecimal Numbers are represented in Sign Magnitude Form.

- Most significant bit represents the sign of the number.
  - ➢ 0 indicates a positive number
  - ➢ 1 indicates a negative number

DIV SOURCE $_{8/16}$

This is an instruction that is used to divide an unsigned word (16 bits) by a byte (8 bit) or to divide an unsigned double word (32 bits) by a word (16 bit).

a) Word by Byte Division:

- The 16 bit Word must be in the AX register.
- The 8 bit Divisor must be loaded in an 8 bit register or a memory location.
- After division,
  AL ← 8 bit Quotient
  AH← 8 bit Remainder

[If an attempt is made to divide by 0 or if the quotient is too large to fit in the AL register (greater than FFH), then, 8086 does a type 0 Interrupt].

```
Example:
MOV AX, 0009H
MOV BL, 04H
DIV BL

AL=02H Quotient
AH=01H Remainder
```

b) Double Word by Word Division:

- A 32bit word must be loaded in the DX [MS Word] and AX [LS Word] registers respectively.
- Divisor (16 bit) must be loaded in a 16 bit register or a memory location.
  After division
  DX ← 16 bit Remainder
  AX ← 16 bit Quotient

```
Example
MOV DS, 0000H
MOV AX, 0008H
MOV BX, 0003H
DIV BX
AX ← 0002H Quotient
DX←0002H Remainder
```

When the source and destination are of different data size

- In some programs user needs to use **CBW** to convert byte to word and
- **CWD** convert word to double word

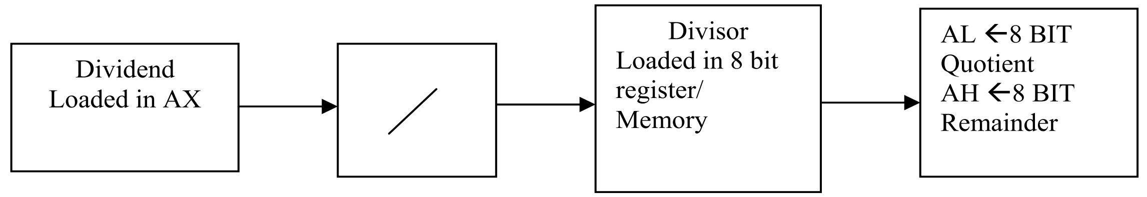


Figure: 11.1: Concept Diagram for 16 Bit By 8 Bit Division:

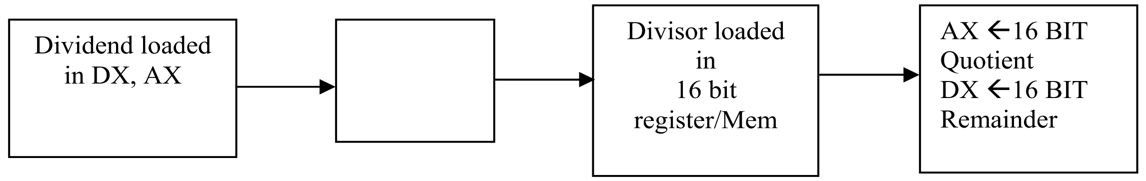


Figure 11.2: Concept Diagram for 32 Bit by 16 Bit Division:

IDIV SOURCE$_{8/16}$

This instruction is used to divide a signed word by a signed byte or to divide a signed double word (32 bits) by a signed word.

Examples: IDIV BL
IDIV BX

VIII. Algorithm

Write algorithm for Division of 8 bit by 8 bit no.

………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………….

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………..

IX. Flow Chart (Student shall draw the flow chart)

IX. Assembly Language Program Code

Program for16 bit by 8bit division.

```
DATA SEGMENT
          NUMBER1  DW  0009H
          NUMBER2  DB  02H
         Quotient  DB 1 DUP(0)
         Remainder DB 1 DUP(0)
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START:MOV DX  ,   DATA
         MOV   DS ,DX
         MOV   AX ,NUMBER1
         MOV   BL ,NUMBER2
         DIV BL
         MOV   Quotient, AL
         MOV  Remainder, AH
       MOV AH , 4CH
         INT 21H
         CODE  ENDS
          END START
```

X. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0		

XI Precautions

1. If overflow error occurs necessary instruction like CBW,CWD can be used for accumulating result by extending sign magnitude of data variable.
2. Collect the result of Division in appropriate variable.
3. Signed result of division shall be verified properly.
4. Keep update of flags after division.

XII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. Complete the given tables:

Table 11.1: For 16 bit by 8 bit division

	CASE I	CASE II	CASE III
NUMBER1	0042H		
NUMBER2	15H		
DIVISION RESULT			

Table 11.2: For 16 bit by 16 bit division

	CASE I	CASE II	CASE III
NUMBER1	0042H		
NUMBER2	15H		
DIVISION RESULT			

2. Elaborate result of multiplication for 8 bit and 16 bit multiplication in terms of AH,AL, DX,AX respectively with proper example..
3. Write instruction used for signed division?
4. State the addressing mode of IDIV instruction.
5. 4321H/37H=……………….Remainder=………….
6. DIV instruction can be used for Division of 8/8 bit, and ……… Numbers.(32/8,16/8,20/8,16/16,32/32,24/16,8/16,8/24)
7. State the name of the register for the storage of Quotient after 16/8 bit Quotient after 32/16 bit Division.
8. State the name of the register for the storage of Remainder after 16/8 bit Division.
9. In 2's Complement form, -7 is represented as ……….

(Space for answers)

XVII Exercise

Attempt Q1. and teacher shall allot any two out of Q………, Q…….. and Q……… the following:

1. Complete following table:

Table 11.3:

Sr. No.	Errors Occurred	Error corrected as	Comment on error
1	DIV AX,BX	DIV BX or DIV AX	Gives error during assembly
2	IDIV (BX)		
3	DIV AL,BX		
4	IDIV CX		
5	DIV AL,BX		
6			

2. Draw a flowchart for 8 bit by 8 bit division.
3. Write assembly language program for division of16 bit by 16 bit no.
4. Complete given data using program developed in laboratory:
5. Flowchart for16 bit by 8bit signed division.
6. Observe the errors and write the corrected statement

Table 11.4:

Statements with Errors are	Corrected Statements
1. NUMBER1 DB 0009H	
2. MOV Quotient, AH	
3. Remainder DW 1DUP(0)	
4. MOV Remainder, A L	

(Space for answers)

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

XVIII References / Suggestions for further Reading

1. https://www.tutorialspoint.com/assembly_programming (As on 19th Jan 2018)
2. https://www.elprocus.com/8086-assembly-language-programs-explanation/ (As on 19th Jan 2018)
3. http://www.pcpolytechnic.com/computer/learning.html (As on 19th Jan 2018)

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

List of Students /Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..
4. …………………………………..

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

Practical No. 12: Binary coded decimal addition / subtraction of BCD Numbers

I. Practical Significance

BCD representation of decimal number system is easy to encode and decode which helps to understand microprocessor based systems.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:**
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 8: Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
4. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

- **Build digital systems including microprocessor based systems**

1. Develop algorithm for implementing BCD Addition /Subtraction.
2. Interpret BCD operation to perform for assembly language programming in 8086.
3. Assemble, link and debug assembly language program.
4. Verify or locate contents of memory in terms of segments, registers, operands, variables, flags before and after execution of the program.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.
ii) Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Write an ALP to add / sub two BCD numbers.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.
3. Follow practices related to environmental context.

VII. Minimum Theoretical Background

Binary-coded Decimal or **BCD** is a way of representing a decimal number as a string of bits suitable for use in electronic systems. Rather than converting the whole number into binary, BCD splits the number up into its digits and converts each digit to 4-bit binary. Thus, for example, 345 becomes 0011 0100 0101

BCD Arithmetic

- It is arithmetic operation.
- Microprocessor performs arithmetic operations on binary, hexadecimal numbers.
- Arithmetic operations cannot performed directly on BCD numbers in Packed BCD format.

Example: Let, 0101 is added with 0110.

					0	1	0	1	BCD1
				+	0	1	1	0	BCD2
					1	0	1	1	Invalid BCD
				+	0	1	1	0	Add 6 (0110)
	0	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**Valid BCD**

- **Check yourself.**

  $(0101)_2 = (5)_{10}$ and $(0110)_2 = (6)_{10}$

  $(5)_{10}+(6)_{10}=(11)_{10}$

	0	0	0	1	0	0	0	1	BCD1
+	0	0	1	**0**	1	0	0	1	BCD2
	0	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**Invalid BCD**
+	0		0	**0**	0	1	1	0	Add 6 (0110)
		1	1	1	1	1			Carry
	0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**Valid BCD**

- This is the process of BCD Addition.

Method of BCD Subtraction:

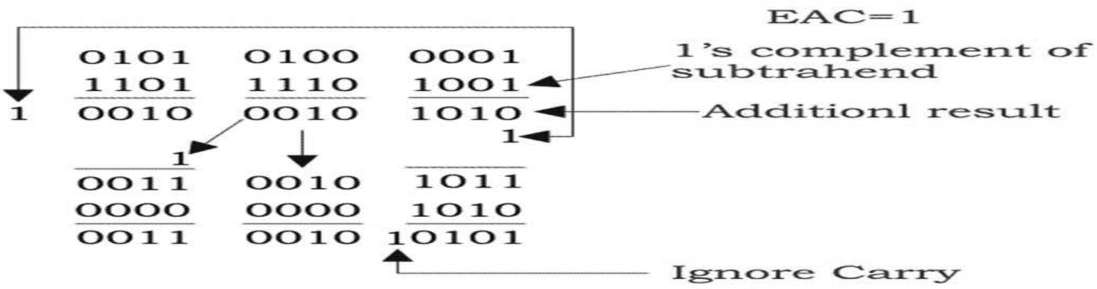
BCD Subtraction (1’s compliment method) There are several steps for this method shown below. They are:-

1. Perform 1’s compliment of the subtrahend..
2. Add complimented subtrahend to the other number from which the subtraction is to be done.
3. Now in BCD Subtraction there is a term ‘**EAC (end-around-carry)**’. If there is a carry i.e if EAC = 1 the result of the subtraction is +ve and if EAC = 0 then the result is –ve. A table shown below gives the rules of EAC.

Table 12.1: Status of End Around Carry

carry of individual groups	EAC = 1	EAC = 0
1	Transfer real result of adder 1 and add 0000 in adder 2	Transfer 1's compliment result of adder 1 and add 1010 in adder 2
0	Transfer real result of adder 1 and add 1010 in adder 2	Transfer 1's compliment result of adder 1 and add 0000 to adder 2

4.In the final result if any carry bit occurs the it will be ignored.



$$(0101\ 0100\ 0001) - (0010\ 0001\ 0110) = (0011\ 0010\ 0101)$$

Figure 12.1: Final Result

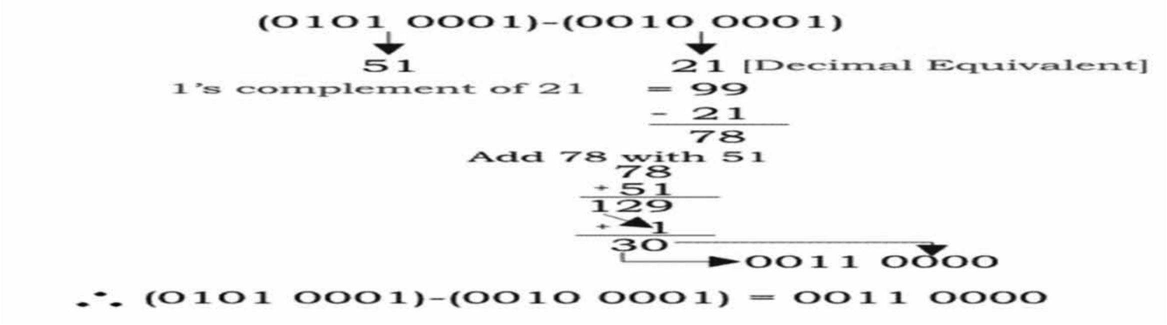
Therefore,

$$(0101\ 0100\ 0001) = (541)_{10}$$
$$(0010\ 0001\ 0110) = 216_{10}$$
$$(0011\ 0010\ 0101) = 325_{10}$$

Figure 12.2: Output

- The decimal result will be changed into **BCD** codes to get the result in BCD. Therefore from the example we can conclude the final result of BCD Subtraction i.e

$$(0101\ 0001)_{BCD} - (0010\ 0001)_{BCD} = (0011\ 0000)_{BCD}$$



Binary Coded Decimal Subtraction using 10’s compliment is same as in case of 9’s compliment, here the only difference is that instead of 9’s compliment we have to do 10’s compliment of the subtrahend.

Sample Calculations:

Table: 12.2: Decimal Adjust Instructions DAA and DAS

Instruction	Description	Example			
		BCD	HEX	BINARY	Stored BCD
DAA	;Decimal Adjust After Addition ✓ If value of lower nibble in Accumulator is greater than 9 or set AF flag then DAA Adds 06H to lower nibble of accumulator **i.e. Lower Nibble of AL>9 or AF =1 then AL =AL +06**. ▪ If value of Higher nibble in Accumulator is greater than 9 or set AF flag then DAA Adds	36 +27 63	36 +27 5D	0011 0110 0010 0111 0101 1101 0000 0110 0110 0011	63
DAS	;Decimal Adjust After Subtraction ✓ If value of lower nibble in Accumulator is greater than 9 or set AF flag then DAA subtracts 06H from lower nibble of accumulator AL **i.e. Lower Nibble of AL>9 or AF =1 then AL =AL - 06**. ▪ If value of Higher nibble in Accumulator is greater than 9 or set CF flag then DAA subtracts 06H from Upper nibble of accumulator AL **Higher nibble of AL >9 or CF =1 then AL =AL - 60.**	35 -18 53	35 -1B 1A	0011 0101 0001 1011 0010 0010 0000 0110 --------------	53

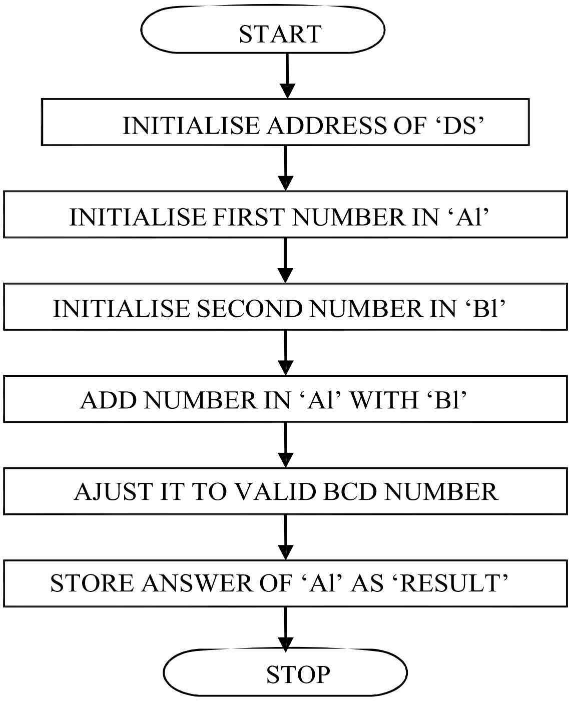
VIII. Algorithm

Algorithm for program to add / Sub, Two BCD numbers in order to understand BCD arithmetic using procedure.

1. Initialize the data segment with numbers on which BCD operations to be performed.
2. Initialize necessary supporting variable to store results generated during above different operations concern to BCD arithmetic operation.
3. Store result(s) during every procedure execution in respective variables

IX Flow Chart

- **Flow chart for addition of two 8 bit BCD numbers.**


START
INITIALISE ADDRESS OF 'DS'
INITIALISE FIRST NUMBER IN 'Al'
INITIALISE SECOND NUMBER IN 'Bl'
ADD NUMBER IN 'Al' WITH 'Bl'
AJUST IT TO VALID BCD NUMBER
STORE ANSWER OF 'Al' AS 'RESULT'
STOP

- **Flow chart for Subtraction of two 8 bit BCD numbers.**

X Assembly Language Program Code

	Program 1(a): Addition of 8 bit BCD numbers using procedure
	DATA SEGMENT
	BCD1 DB 04H
	BCD2 DB 06H
	BCD_SUM DB ?
	DATA ENDS
	CODE SEGMENT
START:	MOV AX,DATA
	MOV DS, AX
	MOV al, BCD 1, ADD al, BCD2
	DAA
	MOV BCD_SUM, AX
	MOV AH,4CH
	INT 21H
	CODE ENDS
	END START

XI Resources required

Sr. No.	**Name of Resource**	**Specification**	**Quantity**	**Remarks**
1.	**Hardware: Computer System**	**Computer (i3-i5 preferable), RAM minimum 2 GB and onwards**	**As per batch size**	**For Experiment no. 8 to 16**
2.	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3.	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII Precautions

1. Check that the result stored is in VALID BCD or not?
2. Procedure shall be called at appropriate place.

XIII Resources used

S. No.	**Name of Resource**	**Specification**
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. Complete the given tables:

Table 12.3: For 8 bit BCD addition

	CASE I (8 bit)	**CASE II (16 bit)**
BCD1	48H	
BCD2	19H	
BCD-SUM **Result Before BCD adjust**		
BCD-SUM **Result After BCD addition**		

Table 12.4: For 8 bit BCD Subtraction

	CASE I (8 bit)	**CASE II (16 bit)**
BCD1	89H	
BCD2	15H	
BCD-DIFF **Result Before BCD adjust**		
BCD-DIFF **Result After BCD addition**		

2. Test the Following table 12.5.

Table 12.5:

Label	Mnemonics	1.	Complete the table using given program segment			
			AX	**BCD1**	**BCD2**	**SUM**
	MOV AX, BCD1			8579	9389	
	ADD AX, BCD2					
	DAA					
	MOV SUM,AX	2.	**Complete the table using given program segment**			
			AX	**BCD1**	**BCD2**	**SUM**
				5432	4567	
				4422	6677	

3. Explain following instructions used in the program?

Instruction	**Write its operation**
DAA	
DAS	

4. Write the program segment for given expression:
   RESULT = (NUM1+NUM2) - (NUM3 –NUM4) assume these numbers are BCD numbers.

5. Check contents of code segment and write contents of address from cs:0000H to cs:0020H(Refer PROGRAM.LST file to complete table 12.6)

Table 12.6: contents of following locations

Address	Contents	Address	Contents	Address	Contents	Address	Contents
CS:0000		CS:0009		CS:0011		CS:001A	
CS:0001		CS:000A		CS:0012		CS:001B	
CS:0002		CS:000B		CS:0013		CS:001C	
CS:0003		CS:000C		CS:0014		CS:001D	
CS:0004		CS:000D		CS:0015		CS:001E	
CS:0005		CS:000E		CS:0016		CS:001F	
CS:0006		CS:000F		CS:0017		CS:0020	
CS:0007		CS:000G		CS:0018			
CS:0008		CS:0010		CS:0019			

6. Completer following table contents after executing program:

Types	Register		FLAG Register		
General purpose registers	AX		Carry	C	
	BX		Zero	Z	
	CX		Sign	S	
	DX		Overflow	O	
Index Registers	SI		Parity	P	
	DI		Auxiliary	A	
Base Pointer	BP		Interrupt	I	
Stack Pointer	SP		Direction	D	
Segment Registers	DS				
	ES				
	SS				
	CS				
Instruction Register	IP				

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVII Exercise

Attempt Q1. And teacher shall allot ANY TWO questions Q. 2, Q.3, Q.4, Q.5 from the following:

1. Give applications of BCD numbers..
2. Draw Flow chart for addition of two 16 bit BCD numbers.
3. Write assembly language program for addition of two 16 bit BCD numbers.
4. Draw Flow chart for subtraction of two 16 bit BCD numbers.
5. Write assembly language program for subtraction of two 16 bit BCD numbers.

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. https://ncalculators.com/digital-computation/binary-addition-calculator.htm
2. http://www.convertforfree.com/bcd-calculator/
3. http://service.scs.carleton.ca/sivarama/asm_book_web/Instructor_copies/ch11_bcd.pdf
4. https://community.computingatschool.org.uk/files/8797/original.ppt

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

List of Students /Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..

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

Practical No. 13: Binary coded decimal multiplication/ division of BCD numbers

I. Practical Significance

BCD representation of decimal number system is easy to encode and decode which helps to understand microprocessor based systems.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 8: Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
4. **PO 10:Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expected to develop the following skills in you:

Build digital systems including microprocessor based systems

1. Develop algorithm for implementing BCD Multiplication and Division.
2. Interpret BCD operation to perform for assembly language programming in 8086.
3. Write assembly language program for implementation of engineering operation for building basic digital systems.
4. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.

ii) Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Write an ALP to multiply / Divide two BCD numbers.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.
3. Follow practices related to environmental context.

VII Minimum Theoretical Background

BCD Multiplication: The successive addition method can be used to perform BCD multiplication, using ADD, ADC and DAA instructions.

- i.e. 9 X 3 = 27 (Multiplication) can be obtained by 9+9+9 = 27 using counter register CL for byte or CX for word multiplication.
- One of the numbers is taken as counter other number added.

Example: 1001 multiply by 0011.

					1	0	0	1	M1
				X	0	0	1	1	M2
					1	0	0	1	
	+			1	0	0	1	X	
	+		0	**0**	0	0	X	X	
	+	0	0	**0**	0	X	X	X	
	0	0	0	1	1	0	1	1	Valid Binary

This is the process of BCD Multiplication.

BCD Division: The successive subtraction can be used to perform BCD division.

- i.e. 9/3= 3 (Division) can be obtained by 9-3=6 count=1 , 6-3=3 count=2, 3-3=0 count=3 using counter register CL for byte or CX for word division is obtained.

VIII Algorithm

1. To multiply two BCD numbers using successive addition.

1. Initialize the data segment with numbers on which BCD operations to be performed.
2. Store multiplier in M1 and set that as counter register for byte in CL and for word in CX.
3. Initialize result with 0 (zero).
4. Initially count assigned to multiplier.
5. Add the multiplier in result and decrease counter by 1.
6. Perform this till the counter variable M1 is zero.
7. Store result(s) during every iteration execution in result.
8. Adjust the result to BCD using DAA.

2. Algorithm (Division of BCD number using successive subtraction)

1. Initialize data segment.
2. Initialize quotient counter with 0.
3. Initialize result variable with dividend.
4. Result = Result – Divisor.
5. Adjust result to BCD.
6. Increment quotient counter by 1.
7. If result of subtraction >= divisor the perform step 4.
8. Store the remainder and quotient in result
9. Stop.

IX Flow Chart (Student shall draw the flow chart)

1. **Multiplication of two BCD numbers using successive addition.**
2. **Division of BCD number using successive subtraction.**

X Assembly Language Program Code

Program 1(a): Multiplication of two 8 BCD bit multiplication using successive addition.

	CODE SEGMENT
	ASSUME CS:CODE, DS: DATA
START:	MOV AX,DATA
	MOV DS, AX
	MOV CL,M2
	MOV AL,0H
UP:	ADD AL,M1
	DAA
	JNC DOWN
Docon:	INC CARRY
	DEC CL
	JNZ UP
	MOV RESULT,AL
	MOV AH,4CH
	INT 21H
	CODE ENDS
	DATA SEGMENT
	M1 DB 09H
	M2 DB 03H
	RESULT DB ?
	CARRY DB ?
	DATA ENDS
	ENDS START

Program 1(b): Division of two 8 bit BCD no. using successive subtraction.

	CODE SEGMENT
START:	ASSUME CS:CODE, DS: DATA
	MOV AX,DATA
	MOV DS, AX
	MOV AL,DIVIDEND
NEXT:	SUB AL,DIVISOR
	DAS
	INC QUO
	CMP AL,DIVISOR
	JNC NEXT
	MOV REM,AL
	MOV AH,4CH
	INT 21H
	CODE ENDS

	DATA SEGMENT
	DIVISOR DB 03H
	DIVIDEND DB 09H
	QUO DB 0
	REM DB 0
	DATA ENDS
	END START

XI Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII Precautions

1. Check that the result stored is in VALID BCD or not?
2. Confirm the result with manual computation.

XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. Complete the given tables:

For 8 bit by 8 bit/16 bit BCD multiplication .

	CASE I (8 bit)	**CASE II (16 bit)**
M1	07H	1234
M2	10H	6789
BCD-SUM **Result Before BCD adjust**		
BCD-SUM **Result After BCD addition**		

For 8 bit by 8 bit/ 16 bit BCD division.

	CASE I (8 bit)	**CASE II (16 bit)**
BCD1	89H	
BCD2	15H	
BCD-DIFF **Result Before BCD adjust**		
BCD-DIFF **Result After BCD addition**		

2. Explain following instructions used in the program?

Instruction	**Write its operation**
JNZ	
JNC	
DAA	
DAS	

3. Write Algorithm and Assembly Language Program for any one of the following:
   - Perform Division of two 16 BCD bit multiplication using successive addition.
   - Perform Division of two16 BCD bit Division using successive subtraction.

4. Check contents of code segment and Data Segment to complete this table.

Address	Contents	Address	Contents		Address	Contents	Address	Contents
CS:0000		CS:0009			CS:0011		CS:001A	
CS:0001		CS:000A			CS:0012		CS:001B	
CS:0002		CS:000B			CS:0013		CS:001C	
CS:0003		CS:000C			CS:0014		CS:001D	
CS:0004		CS:000D			CS:0015		CS:001E	
CS:0005		CS:000E			CS:0016		CS:001F	
CS:0006		CS:000F			CS:0017		CS:0020	
CS:0007		CS:000G			CS:0018			
CS:0008		CS:0010			CS:0019			

Address	Contents	Address	Contents
DS:0000		DS:0009	
DS:0001		DS:000A	
DS:0002			
DS:0003			
DS:0004			
DS:0005			
DS:0006			
DS:0007			
DS:0008			

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVII Exercise

Attempt Q1. And teacher shall allot Q. 2/Q.3 from the following:

1. Draw flow chart for 16 bit BCD multiplication using successive addition.
2. Write algorithm for BCD multiplication using successive addition.
3. Draw flow chat for BCD division using successive subtraction.

(Space for answers)

....

....

XVIII References / Suggestions for further Reading

1. http://www.convertforfree.com/bcd-calculator/
2. https://ncalculators.com/digital-computation/binary-multiplication-calculator.htm
3. https://ncalculators.com/digital-computation/binary-division-calculator.htm
4. https://community.computingatschool.org.uk/files/8797/original.ppt

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		60%
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		40%
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	100 %

List of Students /Team Members

1. ……………………………………..
2. ……………………………………..
3. ……………………………………..
4. ……………………………………..

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

Practical No. 14: Find Smallest Number from an Array.

I. Practical Significance

To perform lots of repetitive tasks, efficiently and to automate the task such programs are helpful.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2: Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 7: Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of practice in the field of Information Technology.
4. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expected to develop the following skills in you:

- **Build digital systems including microprocessor based systems**

1. Develop algorithm for finding smallest element from given data.
2. Interpret decision making to perform for assembly language programming in 8086.
3. Write assembly language program for implementation of engineering operation for building basic digital systems.
4. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

1. Use registers and instructions of 8086.
2. Develop assembly language programs using 8086.

V. Practical Outcome (POs)

- Develop an ALP to find smallest number from array of n numbers.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.

VII. Minimum Theoretical Background

The numbers N1, N2, N3,….. are compared in such an order, where the first comparison checks if N1 smallest. If so, prints it out and exit; if not so, it omits N1 from further comparison and checks with N2, N3,.. till last number till the smaller number identified.

Types of Searching:

- Process of finding the required/specified element from a set of data or an Array i.e. Smallest element, Largest element, select particular element, Even or Odd element

Finding smallest Element:

- The number which is smallest in value given in an array/ set of data.

```
Let arbitrary numbers x, y,
z smallest =x
If(y < smallest then
        smallest
            =y
If (z <smallest)
        then smallest = z
```

<u>**Sequence**</u>

- An algorithm, and eventually a program,
- is a sequence of instructions,
- It can be a simple instruction or either of the other two constructs.

<u>**Decision**</u>

- Situation to test a condition. If the result of testing is true,
- Follow a sequence of instructions: if it is false,
- Follow a different sequence of instructions.

<u>**Repetition**</u>

- The same sequence of instructions must be repeated.
- To handle this with the repetition or *loop* construct.
- To find the largest integer among a set of integers can use a construct of this kind.

Basic for finding smallest and largest among an array of 'n' numbers.

To find and return the smallest no. among the 'n' numbers

```
{
        Smallest = first no.
        Counter = 1
While (counter <= n)
{
       Current = next no.
       If (current < smallest)
       Smallest = current
       Counter = counter +1
}
Return smallest
}
```

Searching:

- The process of finding the location of a target among a list of objects.
- In the case of a list, searching means that given a value,
- To find the location of the first element in the list that contains that value.

There are two basic searches for lists: sequential search and binary search.

Sequential search:

- Sequential search can be used to locate an item in any list.
- Sequential search is used if the list to be searched is not ordered.
- This technique is generally used only for small lists, or lists that are not searched often.
- In other cases, the best approaches are to first sort the list and then search it using the binary search.
- In a sequential search, start searching for the target from the beginning of the list, which will be continued until to find the target or reach the end of the list

Binary search: binary search requires the list first to be sorted.

The CMP instruction:

- It is an instruction used to compare two numeric data fields, one or both of which are contained in a register or memory or immediate.

LABLE:	CMP	Register /memory, register/memory/immediate

- This instruction is also used with string of characters
- This instruction affects AF, CF, OF, PF, SF and ZF flags. i.e.

```
          CMP DX,00      ; DX= zero ?
          JE NEXT          ; If yes, Jump to
               NEXT (It actions if nonzero)
          …
NEXT:……………..      ; Jump point if DX = zero
```

For instruction CMP CX,BX

Updatation of flags as follows:

Condition	CF	ZF	SF	Effect of instruction
CX=BX	0	1	0	; RESULT OF SUBTRACTION IS 0
CX > BX	**0**	**0**	**0**	**;NO BORROW REQUIRED CF = 0**
CX < BX	1	0	1	;SUBTRACTION REQUIRE BORROW , CF = 1

VIII. Algorithm

- **For finding Smallest no. from an array**:

No.1	**No.2**	**No.3**	**….**	**….**	**….**	**No. N**

Input Data

Step 1	Set Smallest to First no.
Step 2	If second no. is Smallest , set smallest to second no.
Step 3	If the third no. is smallest, set smallest to third no.
	…..
	…..
Step last	If the last no is smallest, set smallest to last no.

Output is Smallest no.

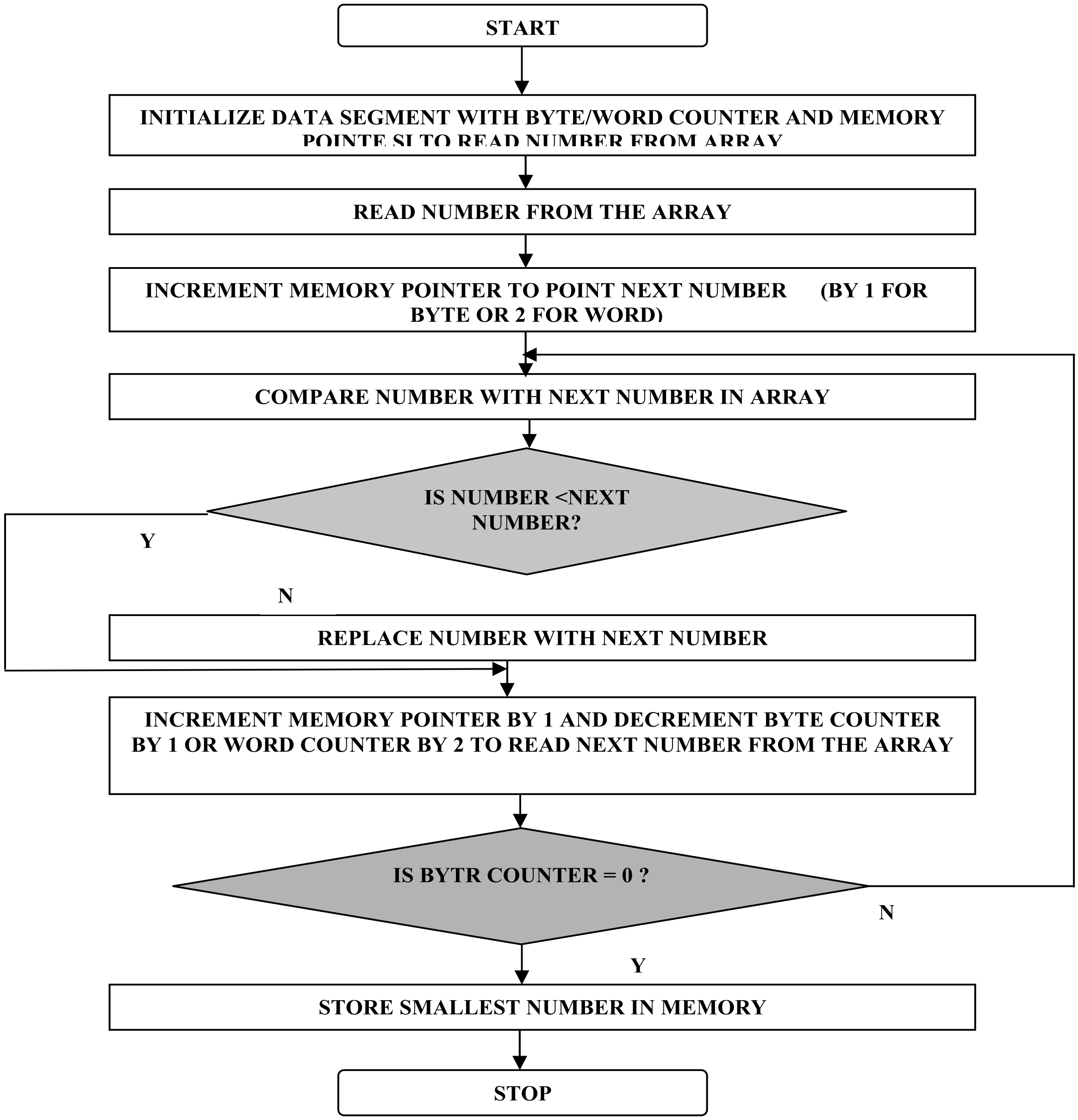
Problem Statement:	
Program 1(a): Algorithm for finding smallest 8-bit no. in an array of data	
1	Initialize data segment , byte counter, and memory pointer to read members from array
2	Read no. from array
3	Increment memory pointer to read next number
4	Decrement byte counter
5	Compare two numbers , IF number < next number then perform step no. 7
6	Replace number with next number which is smallest
7	Increment memory pointer to read next number from array.
8	Decrement counter by 1.
9	If byte counter is not equal then perform step ….5
10	Store smallest no.

(For an ARRAY having five 8 bit numbers)

ARRAY		Step 1		Step 2		Step 3		Step 4
59		AL =59 Compare 59 and 43 Replace AL by smallest no. AL = 43		AL = 43 Compare 43 and 98 No replace as 43 is already smallest no. till now		AL = 43 Compare 43 and 35 Replace AL by 35 the smallest no. till now		AL = 35 Compare 35 and 74 No replace as 35 is already smallest no. till now
43								
98								
35								
74								

IX. Flow Chart

FLOWCHART FOR FINDING SMALLEST NUMBER IN THE ARRAY


START
INITIALIZE DATA SEGMENT WITH BYTE/WORD COUNTER AND MEMORY POINTE SI TO READ NUMBER FROM ARRAY
READ NUMBER FROM THE ARRAY
INCREMENT MEMORY POINTER TO POINT NEXT NUMBER (BY 1 FOR BYTE OR 2 FOR WORD)
COMPARE NUMBER WITH NEXT NUMBER IN ARRAY
IS NUMBER <NEXT NUMBER?
Y
N
REPLACE NUMBER WITH NEXT NUMBER
INCREMENT MEMORY POINTER BY 1 AND DECREMENT BYTE COUNTER BY 1 OR WORD COUNTER BY 2 TO READ NEXT NUMBER FROM THE ARRAY
IS BYTR COUNTER = 0 ?
N
Y
STORE SMALLEST NUMBER IN MEMORY
STOP

X. Assembly Language Program Code

<table>
<tr><th colspan="3">Program for finding smallest number among the SIX 8 bit numbers in the array:</th></tr>
<tr><td></td><td>CODE SEGMENT</td><td>; START OF CODE SEGMENT</td></tr>
<tr><td></td><td>ASSUME CS: CODE, DS: DATA</td><td></td></tr>
<tr><td></td><td>MOV DX, DATA</td><td>;INITIALIZE DATA SEGMENT</td></tr>
<tr><td></td><td>MOV DS, DX</td><td></td></tr>
<tr><td></td><td>MOV CX,05H</td><td>; INITIALIZE BYTE COUNTER TO READ NUMBERS FROM ARRAY (n-1)</td></tr>
<tr><td></td><td>MOV SI,OFFSET ARRAY</td><td>; INITIALIZE MEMORY POINTER TO READ NUMBER</td></tr>
<tr><td></td><td>MOV AL,[SI]</td><td>;READ NUMBER FROM THE ARRAY</td></tr>
<tr><td></td><td>DEC CX</td><td>;DECREMENT COUNTER BY 1</td></tr>
<tr><td>UP:</td><td>INC SI</td><td>;INCREMENT MEMORY POINTER TO POINT NEXT NUMBER IN ARRAY</td></tr>
<tr><td></td><td>CMP AL,[SI]</td><td>;COMPARE NUMBER TO FIND SMALLEST NUMBER</td></tr>
<tr><td></td><td>JC NEXT</td><td>;IF FIRST NUMBER LESS THAN SECOND GO TO UP</td></tr>
<tr><td></td><td>MOV AL,[SI]</td><td>;COMPARE IT WITH NEXT NUMBER</td></tr>
<tr><td>NEXT:</td><td>LOOP UP</td><td>; DECREMENT BYTE COUNTER IF IT IS NON ZERO, COMPARE WITH NEXT NUMBER IN ARRAY</td></tr>
<tr><td></td><td>MOV SMALLEST,AL</td><td>;STORE SMALLEST NUMBER IN MEMORY</td></tr>
<tr><td></td><td>MOV AX, 4C00H</td><td></td></tr>
<tr><td></td><td>INT 21H</td><td></td></tr>
<tr><td></td><td>CODE ENDS</td><td></td></tr>
<tr><td></td><td>DATA SEGMENT</td><td></td></tr>
<tr><td></td><td colspan="2">ARRAY DB 15H,45H,08H,96H,56H,78H,</td></tr>
<tr><td></td><td>SMALLEST DB 00H</td><td></td></tr>
<tr><td></td><td>DATA ENDS</td><td></td></tr>
<tr><td></td><td>END</td><td>;END OF PROGRAM</td></tr>
</table>

Program for finding smallest number among the FOUR 16 bit numbers in the array:	

XI. Resources required

Sr No	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII. Precautions

1. Counter initialized should be one less than the total numbers present.
2. Need to swap when number is smaller.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. Instruction used to decide the smallest number is:……………..
2. Mention no. of swaps (changes) are performed for finding smallest number from the following ?
   Assume (ARRAY DB 04H, 02H, 09H, 10H, 06H)
3. Write an algorithm and Assembly Language Program for any one of the following:
   - Write the steps to find the smallest number in array of three numbers.
   - Write algorithm to count given number in an array
4. Explain instruction MOV AL,[SI]
5. Write flags are affected for the following comparisons of numbers?

NO1	NO2	Using CMP NO1,NO2			NO1	NO2	Using CMP NO2,NO1		
		Carry Flag	Zero Flag	Sign Flag			Carry Flag	Zero Flag	Sign Flag
59	29				59	29			
34	76				34	76			
132	36				132	36			

(Space for answers)

XVII Exercise

Attempt Q1. and teacher shall allot any TWO questions Q. 2/Q.3/Q.4/Q.5 from the following:

1. Student complete the steps

ARRAY	Steps to find Smallest will be
92	
39	
11	
75	
43	
55	

2. Complete the give table of CMP instruction: **CMP BX, CX**
   If CX = 7300H, BX = 2000H
   Complete the given table for above program using your own data:

Condition	CF	ZF	SF	Effect of instruction
CX=BX				
CX > BX				
CX < BX				

3. Given numbers are as follows:
   (Complete the table for finding smallest)
   Use extra columns if needed

ARRAY	Step 1	Step 2	Step 3	Step 4				
7800H								
5064H								
8190H								
2340H								
9000H								

4. Given numbers are as follows: (Complete the table for finding Largest)

ARRAY	Step 1	Step 2	Step 3	Step 4
3000H				
4506H				
7890H				
1234H				
6999H				

5. Write Algorithm to Find largest salary per day for An EXCEL International Ltd. assume of FOUR digit salary per day.)

(Space for answers)

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

....

....

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. http://improvec.blogspot.in/2010/12/program-that-implements-bubble-sort.html
2. http://improvec.blogspot.in/2010/12/program-to-perform-binary- search.html
3. http://improvec.blogspot.in/2010/12/program-to-perform-linear-search.html

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

List of Students /Team Members

1. ……………………………………..
2. ……………………………………..
3. ……………………………………..
4. ……………………………………...

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

Practical No. 15: Find largest number from an array.

I. Practical Significance

To perform lots of repetitive tasks, efficiently and to automate the task such programs are helpful.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 2: Discipline knowledge:** Apply Information Technology knowledge to solve broad- based Information Technology related problems.
3. **PO 8. Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
4. **PO 10: Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

III. Competency and Practical skills

This practical is expected to develop the following skills in you:

- **Build digital systems including microprocessor based systems**

1. Develop algorithm for finding largest element from given data.
2. Interpret decision making to perform for assembly language programming in 8086.
3. Write assembly language program for implementation of engineering operation for building basic digital systems.
4. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

1. Use registers and instructions of 8086.
2. Develop assembly language programs using 8086.

V. Practical Outcome (POs)

Develop an ALP to find largest number from array of n numbers

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.

VII Minimum Theoretical Background

The numbers N1, N2, N3,….. are compared in such an order, where the first comparison checks if N1 largest. If so, prints it out and exit; if not so, it omits N1 from further comparison and checks with N2, N3, …. till last number till the larger number identified.

Types of Searching:

- Process of finding the required/specified element from a set of data or an Array i.e. Smallest element, Largest element, select particular element, Even or Odd element

Finding largest Element:

- The number which is largest in value given in an array/ set of data.
  Let arbitrary numbers x, y, z

```
largest =x
If(y > largest
        then
        largest =y
If (z > largest) then
        largest = z
```

To find and return the largest no. among the 'n' numbers

```
{
        Largest = first no.
        Counter = 1
While (counter <= n)
{
          Current = next no.
        If (current > largest)
        largest = current
        Counter = counter +1
}
Return largest
}
```

Searching:

- The process of finding the location of a target among a list of objects.
- In the case of a list, searching means that given a value,
- To find the location of the first element in the list that contains that value.

There are two basic searches for lists: sequential search and binary search.

Sequential search:

- **In a sequential search,** start searching for the target from the beginning of the list, which will be continued until to find the target or reach the end of the list
- **Binary search:** binary search requires the list first to be sorted.

The CMP instruction:

- It is an instruction used to compare two numeric data fields, one or both of which are contained in a register or memory or immediate.

LABLE:	CMP	Register /memory, register/memory/immediate

- This instruction is also used with string of characters
- This instruction affects AF, CF, OF, PF, SF and ZF flags.

Condition	CF	ZF	SF	Effect of instruction
CX=BX	0	1	0	; RESULT OF SUBTRACTION IS 0
CX > BX	0	0	0	;NO BORROW REQUIRED CF = 0
CX < BX	**1**	**0**	**1**	**;SUBTRACTION REQUIRE BORROW , CF = 1**

VIII Algorithm

- **Procedure for finding Largest no. from an array**:

No.1	**No.2**	**No.3**	**….**	**….**	**….**	**No. N**	**Input Data**

Step 1	Set Largest to First no.
Step 2	If second no. is Largest , set largest to second no.
Step 3	If the third no. is largest, set largest to third no.
	…..
	…..
Step last	If the last no is largest, set largest to last no.

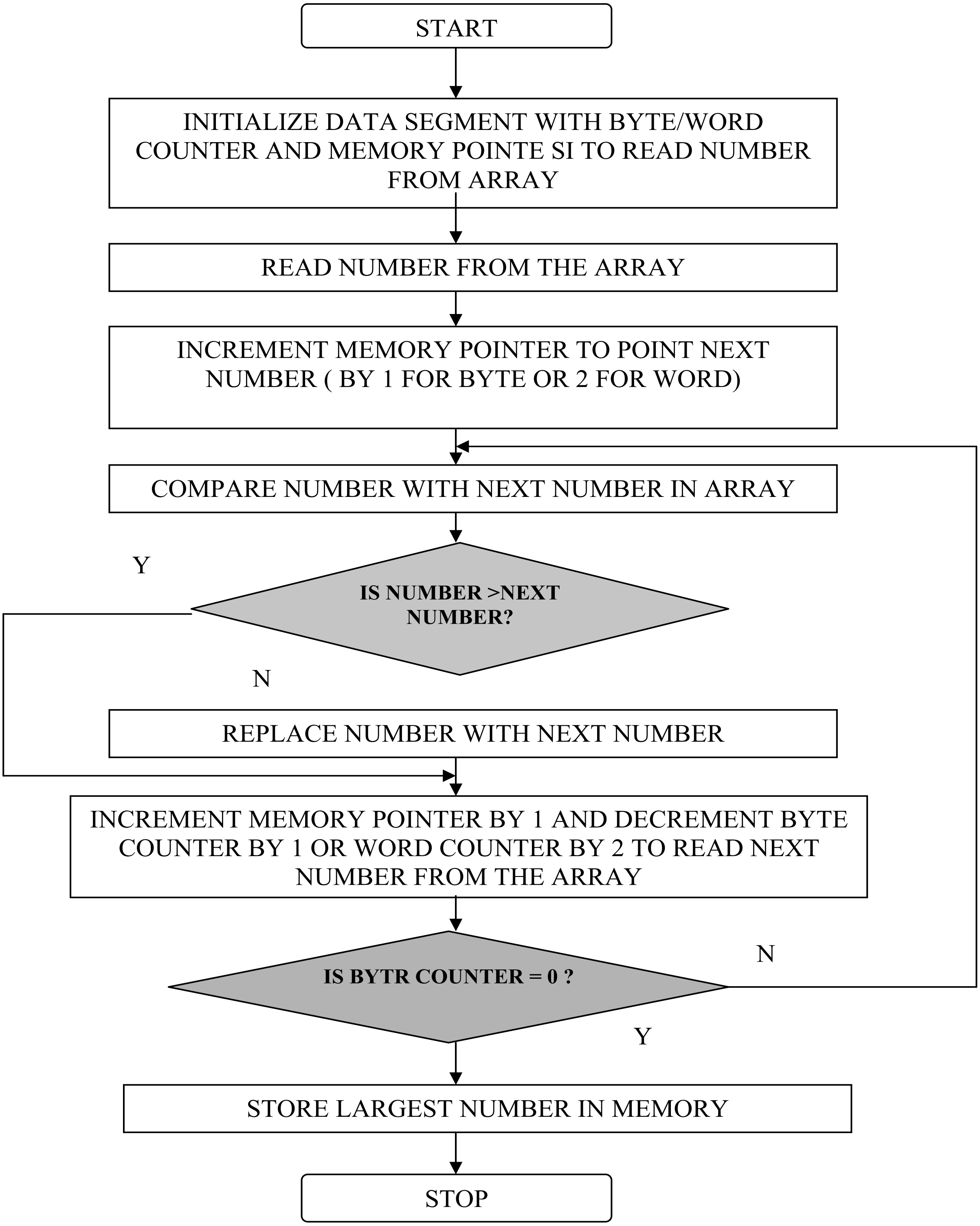
Output is Largest no.

This is one of ways to solve problem. This algorithm receives a list of n numbers as input and gives the largest number as output.

Problem Statement:

<table>
<tr><th colspan="2"><b>Program 1(a): Algorithm for finding Largest from 8-bit no. in an array of data</b></th></tr>
<tr><td>1</td><td>Initialize data segment , byte counter, and memory pointer to read members from array</td></tr>
<tr><td>2</td><td>Read no. from array</td></tr>
<tr><td>3</td><td>Increment memory pointer to read next number</td></tr>
<tr><td>4</td><td>Decrement byte counter</td></tr>
<tr><td>5</td><td>Compare two numbers ,
IF number > next number then perform step no. 7</td></tr>
<tr><td>6</td><td>Replace number with next number which is largest</td></tr>
<tr><td>7</td><td>Increment memory pointer to read next number from array.</td></tr>
<tr><td>8</td><td>Decrement counter by 1.</td></tr>
<tr><td>9</td><td>If byte counter is not equal then perform step ….5</td></tr>
<tr><td>10</td><td>Store largest no.</td></tr>
</table>

IX Flow Chart


START
INITIALIZE DATA SEGMENT WITH BYTE/WORD COUNTER AND MEMORY POINTE SI TO READ NUMBER FROM ARRAY
READ NUMBER FROM THE ARRAY
INCREMENT MEMORY POINTER TO POINT NEXT NUMBER (BY 1 FOR BYTE OR 2 FOR WORD)
COMPARE NUMBER WITH NEXT NUMBER IN ARRAY
Y
IS NUMBER >NEXT NUMBER?
N
REPLACE NUMBER WITH NEXT NUMBER
INCREMENT MEMORY POINTER BY 1 AND DECREMENT BYTE COUNTER BY 1 OR WORD COUNTER BY 2 TO READ NEXT NUMBER FROM THE ARRAY
N
IS BYTR COUNTER = 0 ?
Y
STORE LARGEST NUMBER IN MEMORY
STOP

X Assembly Language Program Code

Program for finding largest number among the five 8 bit numbers in the Array:		
	CODE SEGMENT	; START OF CODE SEGMENT
	ASSUME CS: CODE, DS: DATA	
	MOV DX, DATA	;INITIALIZE DATA SEGMENT
	MOV DS, DX	
	MOV CX,04H	; INITIALIZE BYTE COUNTER TO READ NUMBERS FROM ARRAY (n-1)
	MOV SI,OFFSET ARRAY	; INITIALIZE MEMORY POINTER TO READ NUMBER
	MOV AL,[SI]	;READ NUMBER FROM THE ARRAY
UP:	INC SI	;INCREMENT MEMORY POINTER TO POINT NEXT NUMBER IN ARRAY
	CMP AL,[SI]	;COMPARE NUMBER TO FIND LARGEST NUMBER
	JNC NEXT	;IF FIRST NUMBER GREATER THAN SECOND GO TO UP
	MOV AL,[SI]	;COMPARE IT WITH NEXT NUMBER
NEXT:	LOOP UP	; DECREMENT BYTE COUNTER IF IT IS NON ZERO, COMPARE WITH NEXT NUMBER IN ARRAY
	MOV LARGEST,AL	;STORE LARGEST NUMBER IN MEMORY
	MOV AX, 4C00H	
	INT 21H	
	CODE ENDS	
	DATA SEGMENT	
	ARRAY DB 15H,45H,08H,56H,78H	
	LARGEST DB 00H	
	DATA ENDS	
	END	;END OF PROGRAM

XI Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII Precautions

1. Counter initialized should be one less than the total numbers present.
2. Need to swap when number when larger.

XII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. Instruction used to decide the largest number is:……………..
2. Mention no. of swaps (changes) are performed for finding smallest number from the following ?

Assume (ARRAY DB 39H, 32H, 09H, 91H, 46H

3. Write a program Algorithm and Assembly Language Program for **ANY ONE** of the following:
   - Give steps to find the Largest number in array of three numbers.
   - Write algorithm to count given number in an array
4. Explain instruction MOV [SI], AL
5. Mention flags are affected for the following comparisons of numbers?

NO1	**NO2**	**Using CMP NO1,NO2**			**NO1**	**NO2**	**Using CMP NO2,NO1**		
		Carry Flag	**Zero Flag**	**Sign Flag**			**Carry Flag**	**Zero Flag**	**Sign Flag**
54	67				54	67			
89	49				89	49			
324	556				324	556			

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVII Exercise

Attempt Q1. and teacher shall allot one question from the Q. 2/Q.3/Q.4 :

1. Student complete the steps

ARRAY	Steps to find Smallest will be
78	
13	
88	
92	
21	
39	

2. Complete the give table of CMP instruction: **CMP BX,CX**
   If CX = 3400H, BX = 1200H

Condition	**CF**	**ZF**	**SF**	**Effect of instruction**
CX=BX				
CX > BX				
CX < BX				

3. Given numbers are as follows:
   (Complete the table for finding largest)
   Use extra columns if needed

ARRAY	**Step 1**	**Step 2**	**Step 3**	**Step 4**				
3000H								
4506H								
7890H								
1234H								
6999H								

4. Write ALP to Find largest salary per day for An EXCEL International Ltd.(assume of FOUR digit salary)

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. http://improvec.blogspot.in/2010/12/program-that-implements-bubble-sort.html
2. http://improvec.blogspot.in/2010/12/program-to-perform-binary- search.html
3. http://improvec.blogspot.in/2010/12/program-to-perform-linear-search.html

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

List of Students /Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..
4. ………………………………….

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

Practical No. 16: Program to perform block transfer

I. Practical Significance

Block transfer helps in improving performance and to allow prefetching of data for processing in microprocessor base system.

II. Relevant Program Outcomes (POs)

1. **PO 1: Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.
2. **PO 3: Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.
3. **PO 7: Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of practice in the field of Information Technology.
4. **PO 8: Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

III. Competency and Practical skills

This practical is expected to develop the following skills in you:

- **Build digital systems including microprocessor based systems**
  1. Develop algorithm for implementing Block transfer of data from one memory location to another memory location.
  2. Interpret Block transfer operation(s) to perform for assembly language programming in 8086.
  3. Write assembly language program for implementation of engineering operation for building basic digital systems.
  4. Assemble, link and debug assembly language program.

IV. Relevant Course Outcome(s)

i) Use registers and instructions of 8086.
ii) Develop assembly language programs using 8086

V. Practical Outcome (POs)

Write an ALP to perform block transfer from one memory location to another.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety measures
2. Follow ethical practices.

VII Minimum Theoretical Background

Non-Overlapped Block Transfer

his refers to transferring the block of data from source memory locations to destination memory locations. Counter is set, whose value is equal to the block length and on each transfer of data from source to destination, the counter is decremented by one and the memory pointer is incremented by one. This process is repeated till the counter becomes zero.

Before Block Transfer				
Source Block			Destination Block	
C001	12	→	D001	15
C002	24	→	D002	30
C003	36	→	D003	45
C004	48	→	D004	60
C005	60		D005	75
C006	72		D006	90

Before Block Transfer				
Source Block			Destination Block	
C001	12		D001	12
C002	24		D002	24
C003	36		D003	36
C004	48		D004	48
C005	60		D005	60
C006	72		D006	72

Overlapped Block Transfer

This refers to exchanging the block of data between one-memory location to other. Here, the counter is set whose value is equal to block length and on each transfer of data from source to destination, the counter is decremented by one and the memory pointer is incremented by one. This process is repeated till counter becomes zero.

Before Block Transfer				
Source Block			Destination Block	
C001	12	↔	D001	15
C002	24	↔	D002	30
C003	36	↔	D003	45
C004	48	↔	D004	60
C005	60		D005	75
C006	72		D006	90

After Block Transfer				
Source Block			Destination Block	
C001	15		D001	12
C002	30		D002	24
C003	45		D003	36
C004	60		D004	48
C005	75		D005	60
C006	90		D006	72

Instruction LEA SI, BLOCK1 and LEA DI, BLOCK2:

This instruction will load the starting address of BLOCK1 into SI and Address of BLOCK2 register into the data segment register.

Instruction INC SI and INC DI:

This instruction will increment the contents of SI and DI registers.

VIII Algorithm

The source block is at address 4000H and Destination Block is at address 5000H. Write an assembly Language program to move block of N bytes from source location to a destination location. (N= 10)

STEP1:

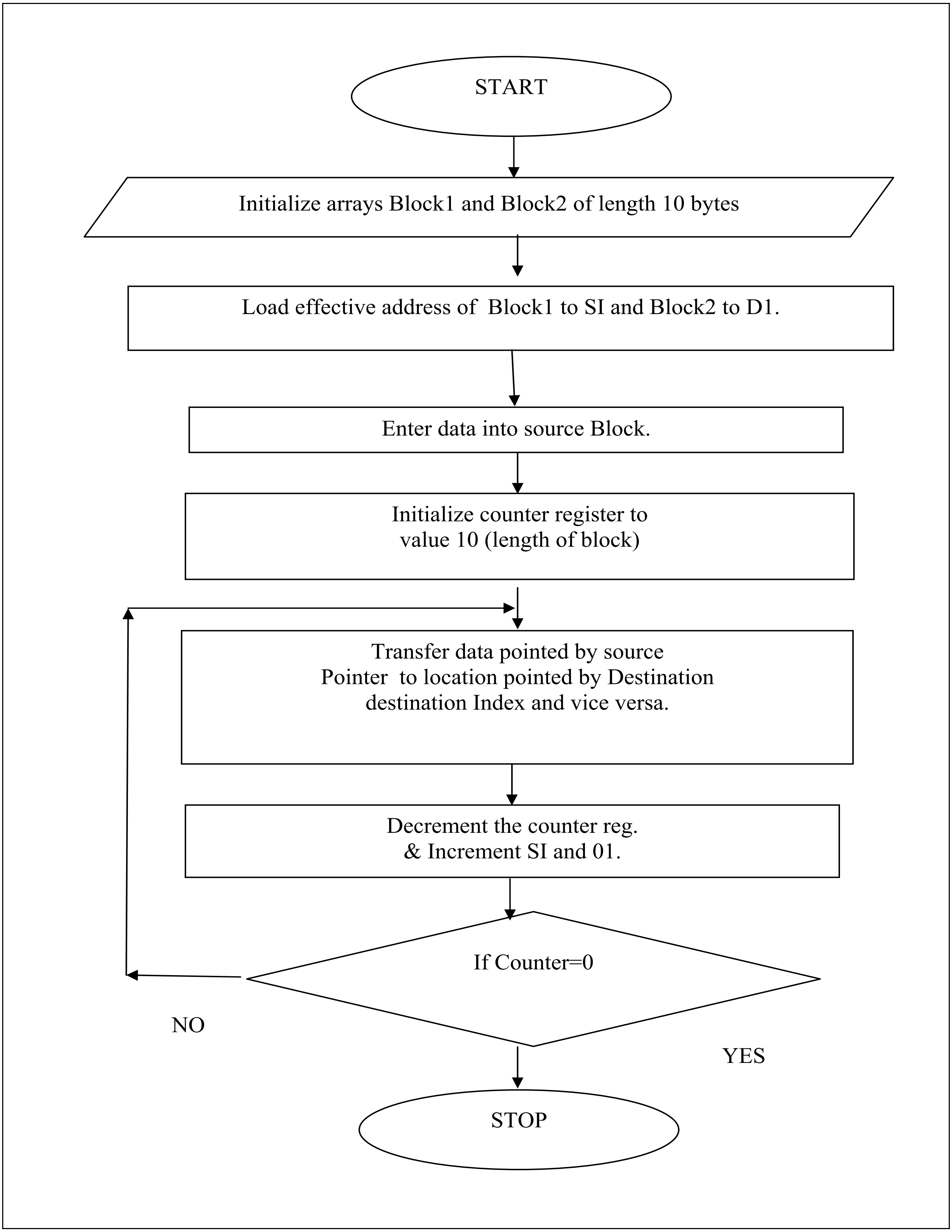
Analyze the given problem and develop the algorithm.

Algorithm for transfer block of data from one memory block to other.

1. Initialize the Data Segment with address of Source and Destination block.
2. Load effective address of BLOCK1 to SI and BLOCK2 to D1.
3. Enter data in to source block.
4. Initialize S1 = Start of destination block
5. Initialize counter = N =10
6. Transfer contents from source location to destination location and vise versa.
7. Decrement the counter.
8. Increment SI and DI.
9. Is count =0? NO- go to step 6
10. Display the contents and stop.

IX Flow Chart

Flow Chart for transfer block of data from one memory block to other.


START
Initialize arrays Block1 and Block2 of length 10 bytes
Load effective address of Block1 to SI and Block2 to D1.
Enter data into source Block.
Initialize counter register to value 10 (length of block)
Transfer data pointed by source Pointer to location pointed by Destination destination Index and vice versa.
Decrement the counter reg. & Increment SI and 01.
If Counter=0
NO
YES
STOP

X Assembly Language Program Code

	Mnemonics	Comments
	DATA SEGMENT	;Start of data segment
	BLOCK1 DB 10 DUP(0)	; Declare array BLOCK1 of size 10 bytes.
	BLOCK2 DB 10DUP(0)	; Declare array BLOCK2 of size 10 bytes.
	DATA ENDS	;End of data segment:
	CODE SEGMENT	;Start of code segment
	ASSUME CS:CODE > S:DATA, ES: DATA	
START:	MOV DX,DATA	;Initialization of Data Segment register
	MOV DS, DX	
	MOV DX,EXTRA	;Initialization of Extra Segment register
	MOV ES,DX	
	LEA SI, BLOCK1	;Load effective address of Block1 to SI
	LEA D1, BLOCK2	;Load effective address of Block2 to 01
	MOV CX, OOO9H	; Initialize the counter to Block length = 10 .
	CLD	;Clear DF
	REP MOVSB	;Move the contents of memory location ;pointed by SI to location pointed by DI
	MOV AH, 4CH	
	INT 21H	
	CODE ENDS	;End of code segment
	END START	

PROGRAM TO COPY ONE ARRAY TO ANOTHER ARRAY in Assembly Language

Write comments for following program segment.

	Mnemonics	Comments
	DATA SEGMENT	
	A DB 1,2,3,4,5,6,7,8,9,10	
	B DB 10 DUP(0)	
	DATA ENDS	
	CODE SEGMENT	
	ASSUME DS:DATA,CS:CODE	
START:	MOV AX,DATA	
	MOV DS,AX	
	MOV CL,10	
	LEA BX,A	
	LEA SI,B	
L1:	MOV CH,BYTE PTR[BX]	
	MOV BYTE PTR[SI],CH	
	MOV DH,BYTE PTR[SI]	
	INC BX	
	INC SI	
	DEC CL	
	CMP CL,00	
	JNZ L1	
	MOV AH,4CH	
	INT 21H	
	CODE ENDS	
	END START	

XI Resources required

Sr. No	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For Experiment no. 8 to 16
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Any Editor like EDIT, Notepad, Turbo Assembler TASM/MASM 3.0, Linker TLINK/LINK 2.0 Debugger TD/Debug		

XII Precautions

1. Block of memory will decide the method of block transfer using or without using string instruction.
2. .Block size shall be known to the programmer.

XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV Result (Output of the Program)

....

....

....

XV Conclusion(s)

....

....

....

XVI Practical Related Questions

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

1. State concept of block transfer of block1 and block 2 using SI and DI registers?
2. What are the use of source Index and Destination Index registers?
3. Explain instruction MOV [BX] ,AX
4. State significance of SI and DI registers in the program?
5. State the use of LOOP BACK statement executed?
6. Write Industrial applications of Block transfer program.
7. List the string instructions.
8. List all string instructions of 8086.Explain each with suitable example.

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVII Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

1. Draw the flow chart showing how the block of N bytes is read into memory.
2. Assembly Language Coding Sheet for block transfer without data using string instructions.
3. Algorithm for transferring block of data without using string instructions

(Space for answers)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVIII References / Suggestions for further Reading

1. http://jntuimplab.blogspot.in/2008/06/block-data-transfer-program-for-8086.html
2. http://jntuimplab.blogspot.in/search/label/8086
3. https://ekendraonline.com/engg/computer-architecture/write-an-assembly-program-to-copy-a-block-of-data-from-one-memory-to-another/
4. https://programsinengineering.blogspot.in/2015/08/to-transfer-block-of-data-using-string.html

XIX Assessment Scheme

Performance indicators		Weightage
Process related:15 Marks		**60%**
1	Handling of the components/IC	20 %
2	Making connections of IC on Breadboard	30 %
3	Working in team to perform experiment	10 %
Product related:10 Marks		**40%**
4	Result & Conclusion	20 %
5	Answers to Practical related questions	15 %
6	Submitting the journal in time	05%
	Total (25 Marks)	**100 %**

List of Students /Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..
4. …………………………………..

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

List Of Laboratory Manuals Developed by MSBTE

First Semester:

1	Fundamentals of ICT	22001
2	English	22101
3	English Work Book	22101
4	Basic Science (Chemistry)	22102
5	Basic Science (Physics)	22102

Second Semester:

1	Bussiness Communication Using Computers	22009
2	Computer Peripherals & Hardware Maintenace	22013
3	Web Page Design with HTML	22014
4	Applied Science (Chemistry)	22202
5	Applied Science (Physics)	22202
6	Applied Machines	22203
7	Basic Surveying	22205
8	Applied Science (Chemistry)	22211
9	Applied Science (Physics)	22211
10	Fundamental of Electrical Engineering	22212
11	Elements of Electronics	22213
12	Elements of Electrical Engineering	22215
13	Basic Electronics	22216
14	‘C’ programming Language	22218
15	Basic Electronics	22225
16	Programming in “C”	22226
17	Fundamentals of Chemical Engineering	22231

Third Semester:

1	Applied Multimedia Techniques	22024
2	Advanced Serveying	22301
3	Highway Engineering	22302
4	Mechanics of Structures	22303
5	Building Construction	22304
6	Concrete Technology	22305
7	Strength Of Materials	22306
8	Automobile Engines	22308
9	Automobile Transmission System	22309
10	Mechanical Operations	22313
11	Technology Of Inorganic Chemicals	22314
12	Object Oriented Programming Using C++	22316
13	Data Structure Using ‘C’	22317
14	Computer Graphics	22318
15	Database Management System	22319
16	Digital Techniques	22320
17	Principles Of Database	22321
18	Digital Techniques & Microprocessor	22323
19	Electrical Circuits	22324
20	Electrical & Electronic Measurment	22325
21	Fundamental Of Power Electronics	22326
22	Electrical Materials & Wiring Practice	22328
23	Applied Electronics	22329
24	Electrical Circuits & Networks	22330
25	Electronic Measurments & Instrumentation	22333
26	Principles Of Electronics Communication	22334
27	Thermal Engineering	22337
28	Engineering Matrology	22342
29	Mechanical Engineering Materials	22343
30	Theory Of Machines	22344

Fourth Semester:

1	Hydraulics	22401
2	Geo Technical Engineering	22404
3	Chemical Process Instrumentation & Control	22407
4	Fluid Flow Operation	22409
5	Technology Of Organic Chemicals	22410
6	Java Programming	22412
7	GUI Application Development Using VB.net	22034
8	Microprocessor	22415
9	Database Managment	22416
10	Electric Motors And Transformers	22418
11	Industrial Measurements	22420
12	Digital Electronics And Microcontroller Applications	22421
13	Linear Integrated Circuits	22423
14	Microcontroller & Applications	22426
15	Basic Power Electronics	22427
16	Digital Communication Systems	22428
17	Mechanical Engineering Measurments	22443
18	Fluid Mechanics and Machinery	22445
19	Fundamentals Of Mechatronics	22048

Fifth Semester:

1	Design of Steel and RCC Structures	22502
2	Public Health Engineering	22504
3	Heat Transfer Operation	22510
4	Environmental Technology	22511
5	Operating Systems	22516
6	Advanced Java Programming	22517
7	Software Testing	22518
8	Control Systems and PLC’s	22531
9	Embedded Systems	22532
10	Mobile and Wireless Communication	22533
11	Industrial Machines	22523
12	Switchgear and Protection	22524
13	Energy Conservation and Audit	22525
14	Power Engineering and Refrigeration	22562
15	Solid Modeling and Additive Manufacturing	22053
16	Guidelines & Assessment Manual for Micro Projects & Industrial Training	22057

Sixth Semester:

1	Solid Modeling	17063
2	Highway Engineering	17602
3	Contracts & Accounts	17603
4	Design of R.C.C. Structures	17604
5	Industrial Fluid Power	17608
6	Design of Machine Elements	17610
7	Automotive Electrical and Electronic Systems	17617
8	Vehicle Systems Maintenance	17618
9	Software Testing	17624
10	Advanced Java Programming	17625
11	Mobile Computing	17632
12	System Programing	17634
13	Testing & Maintenance of Electrical Equipments	17637
14	Power Electronics	17638
15	Illumination Engineering	17639
16	Power System Operation & Control	17643
17	Environmental Technology	17646
18	Mass Transfer Operation	17648
19	Advanced Communication System	17656
20	Mobile Communication	17657
21	Embedded System	17658
22	Process Control System	17663
23	Industrial Automation	17664
24	Industrial Drives	17667
25	Video Engineering	17668
26	Optical Fiber & Mobile Communication	17669
27	Therapeutic Equipment	17671
28	Intensive Care Equipment	17672
29	Medical Imaging Equipment	17673

Pharmacy Lab Manual

First Year:

1	Pharmaceutics - I	0805
2	Pharmaceutical Chemistry - I	0806
3	Pharmacognosy	0807
4	Biochemistry and Clinical Pathology	0808
5	Human Anatomy and Physiology	0809

Second Year:

1	Pharmaceutics - II	0811
2	Pharmaceutical Chemistry - II	0812
3	Pharmacology & Toxicology	0813
4	Hospital and Clinical Pharmacy	0816

HEAD OFFICE

Secretary,
Maharashtra State Board of Technical Education
49, Kherwadi, Bandra (East), Mumbai - 400 051
Maharashtra (INDIA)
Tel: (022)26471255 (5 -lines)
Fax: 022 - 26473980
Email: -secretary@msbte.com
Web -www.msbte.org.in

REGIONAL OFFICES:

MUMBAI

Deputy Secretary (T),
Mumbai Sub-region,
2nd Floor, Govt. Polytechnic Building,
49, Kherwadi, Bandra (East)
Mumbai - 400 051
Phone: 022-26473253 / 54
Fax: 022-26478795
Email: rbtemumbai@msbte.com

NAGPUR

Deputy Secretary (T),
M.S. Board of Technical Education
Regional Office,
Mangalwari Bazar, Sadar, Nagpur - 440 001
Phone: 0712-2564836 / 2562223
Fax: 0712-2560350
Email: rbteng@msbte.com

PUNE

Deputy Secretary (T),
M.S. Board of Technical Education,
Regional Office,
412-E, Bahirat Patil Chowk,
Shivaji Nagar, Pune
Phone: 020-25656994 / 25660319
Fax: 020-25656994
Email: rbtepn@msbte.com

AURANGABAD

Deputy Secretary (T),
M.S. Board of Technical Education,
Regional Office,
Osmanpura, Aurangabad -431 001.
Phone: 0240-2334025 / 2331273
Fax: 0240-2349669
Email: rbteau@msbte.com