

Experiments @ Electronics

Volume II

Digital Electronics Practical Book

Authors

Dr. Dinesh V Kala

Mrs. Bhavana Suri



H.S.R.A PUBLICATIONS

Published by

HSRA Publications 2021

#02, Sri Annapoorneshwari Nilaya, 1st Main,

Byraveshwara Nagar, Laggere,

Bangalore – 560058

Ph- 7892793054

Sales Headquarters – Bangalore

Copyright © AUTHORS 2021

This book has been published with all reasonable efforts taken to make the material error-free after the consent of the respective authors. No part of this book shall be used, reproduced in any manner whatsoever without written permission from the editors, except in the case of brief quotations embodied in critical articles and reviews. The Authors of the respective chapters of this book is solely responsible and liable for its content.

All rights reserved.

No part of this publication may be reproduced, transmitted, or stored in any digital or Electronic form. Also photocopying, recording or otherwise without the prior permission of the editor and publisher is strictly prohibited.

ISBN: 978-93-5506-189-8

First Edition 2021

No. of Pages - 212

₹399.00

Preface

After so many years of teaching, it has been realized that most of the times there is a lot of deviation between the concept mentioned in the theory books and what we observe when we do the experiments in the laboratory. Most of the times, it is observed that a student completes the experiments mechanically, without knowing the origin, principle & concept behind the experiments.

The book on Experiments @ Electronics, Volume II, digital electronics practical book, is the outcome of actual experimentations performed in the laboratories, consequently an ideal practical book has come into existence.

The specialty about the book is, every experiment is preceded by appropriate and sufficient theoretical background. Student can just read the theory behind the experiment, understand the experiment and can perform the experiments himself/ herself.

We are sure of the fact that not only the students, even our teacher colleagues will also enjoy the journey through the book of doing the experiments with different techniques and innovation.

Every care has been taken to eliminate misprints and errors, but it is too much to expect that no misprint or error has crept in and we would be grateful to the readers for bringing the same to our notice. Suggestions for further improvement of the book and enlarging its utility will be gratefully acknowledged.

Acknowledgement

I am thankful to all who has contributed for outcome of this beautiful creation. I am thankful to my parents, my wife, my teachers, my guide and my students. Special thanks to Mrs. Bhavana Suri for not only contributing important chapters but seriously and sincerely editing the whole content of the manuscript.

Thanks a lot, to my students Raveena, Artikrwar, Jabeen, Campal for their valuable contribution from student's perspective. Special thanks to my dear friend Suryakant for always motivating me for expressing my concept in the form of the books. Special thanks to my college management and my principal.

Dr. Dinesh V Kala

First & foremost, praises to the almighty, for his blessings throughout. Because of his blessings, I was able to devote time to my learning and complete the book successfully.

It is a pleasure for me to express my sincere gratitude to Dr. Dinesh V Kala for his complete involvement throughout the preparation of the book. It is solely because of his uninterrupted motivation and invaluable guidance that I was able to conceptualize and eventually bring this book into existence.

I am thankful to my husband for his continuous encouragement and support throughout the journey of my learning. I would like to express my deep and sincere gratitude to my parents and my sister for the love, care and prayers to bring in me, the best of my capacity and capability.

I've gained a whole new perspective in learning and consequently translating the whole concept in the form of book which I am sure will be more delightful for all the learners.

Mrs. Bhavana Suri

CONTENTS

Sr. No.	Chapter	Page No.
1.	Universal Gates & Boolean Algebra	1 -33
1.1	<i>a.</i> To verify the truth tables of NAND and NOR gates <i>b.</i> To configure universal gates as other logic gates and verify	
1.2	To verify De - Morgan's theorem	
1.3	To study Half adder and Full Adder using Logic gates	
1.4	To construct and study addition & subtraction using IC 7483 & IC 7486.	
2.	Digital Electronics	34- 119
2.1.1	To study the working of IC 74151 (8:1 Multiplexer).	
2.1.2	<i>a.</i> To construct a 16:1 multiplexer from two 8:1 MUX (IC 74151) <i>b.</i> To study the working of IC 74150 (16:1 Multiplexer).	
2.2	To study the working of IC 74155 (1:4/8 De-multiplexer).	
2.3	To study the working of IC 74LS148 (8:3 Priority Encoder).	
2.4	To study the working of IC 74LS138 (3:8 Decoder).	
2.5	To study the working of seven segment display IC 74LS47/74LS48.	
2.6.1	To study the working of IC 74244 as a unidirectional buffer.	
2.6.2	To study the working of IC 74245 as an octal transceiver buffer.	
2.7	To study the working of D - Latch IC 74LS373.	
2.8	To study the working of Shift-Register IC 74LS95.	
2.9.1	To study the working of 4 - bit Ripple Counter using IC 7493.	
2.9.2	To study the working of BCD Decade Counter using IC 7490.	
2.9.3	To study the working of Pre - Settable Counter using IC 74193.	
3.	D/A Converter & A/D Converter	119-147

	3.1	To study D/A conversion using Binary Weighted Network.	
	3.2	To study D/A conversion using R - 2R Ladder Network.	
	3.3	To study D/A conversion using 8 - Bit IC based DAC.	
	3.4	To study A/D conversion using 8 - Bit IC based ADC.	
4.		μP 8085: Assembly Language Programming	148-167
	4.1	An Introduction on μ P 8085	
	4.2	μ P 8085 Programming I	
	4.3	μ P 8085 Programming II	
	4.4	Delays & their Applications	
	4.5	PPI 8255A : Ports, Control Word, I/O & BSR Mode	
	4.6	An Interface of μ P 8085, PPI 8255 & DAC 1408 : Waveform Generation	
5.		μC 8051: Assembly Language Programming	168-194
	5.1	An Introduction on μ C 8051	
	5.2	μ C 8051 Programming - I	
	5.3	μ C 8051 Programming - II	
	5.4	Delays & their Applications	
	5.5	An Interface of μ C 8051, PPI 8255 & DAC 1408 : Waveform Generation	
6.		Appendix	195-206
	A	An Application: Interfacing Counter, B C Decoder & 7-Segment Display	
	B	An Introduction to 8-Bit, 8-Channel ADC0808	
	C	Pin Diagrams of all Important ICs	

1

Universal Gates and Boolean Algebra

1.1 KNOWING LOGIC GATES

Logic gates are commonly used in integrated circuits. Commonly used logic gates are TTL and CMOS. TTL or Transistor-Transistor Logic ICs use NPN & PNP type Bipolar Junction Transistors. CMOS or Complementary Metal - Oxide - Silicon ICs are constructed from MOSFET or JFET type Field Effect Transistors. TTL IC's are commonly labeled as the 7400 series while CMOS ICs may often be marked as a 4000 series of chips.

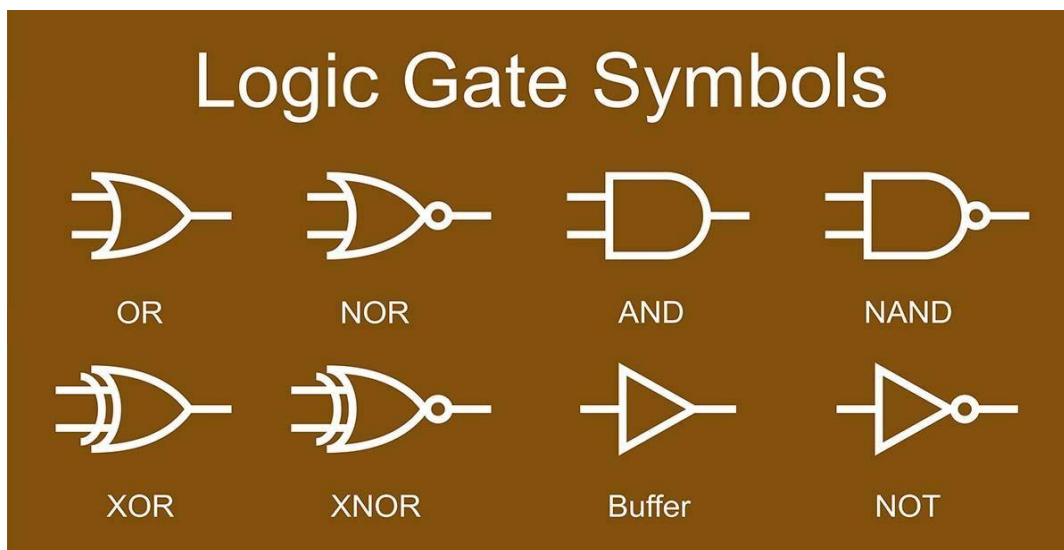


Figure 1.1

1.2 UNDERSTANDING LOGIC GATES

Logic gates are based on Boolean algebra. At any given moment, every terminal is in one of the two binary conditions, *false* or *true*; false represents 0 and true represents 1. In most logic gates, the low state is approximately zero volts (0 V), while the high state is approximately five volts positive (+5 V).

The *AND gate* acts in the same way as the logical "and" operator. The output is "true" when both inputs are "true." Otherwise, the output is "false." In other words, the output is 1 only when both inputs [one] AND [two] are 1. **For example, if a student is to be selected for TYBSC physics to have necessarily [physics] AND [mathematics] in SYBSC level, the AND GATE logic will be implied.**

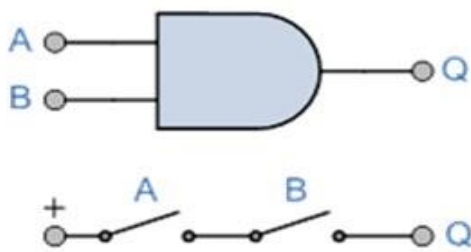


Figure 1.2

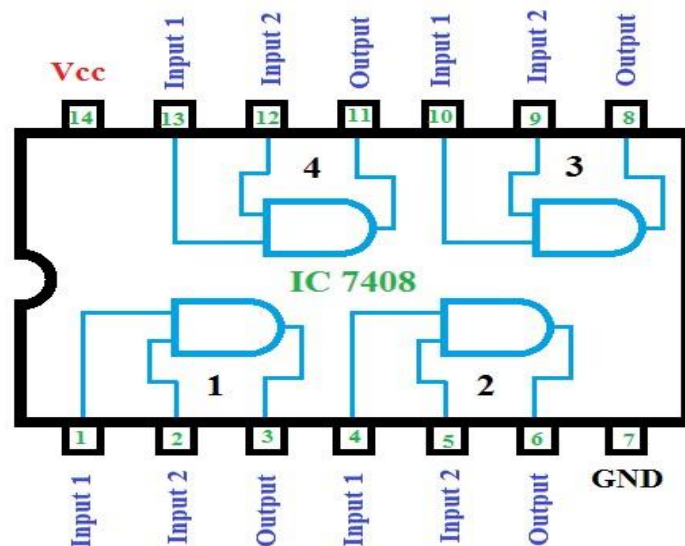


Figure 1.3

Fig. 1.3 shows the pin diagram of IC 7408. It comprises of four AND gates, one V_{CC} and one Ground.

Table 1.1

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

The *OR gate* gets its name from the fact that it responds as the logical of inclusive "or." The output is "true" if either or both inputs are "true." If both inputs are "false," then the output is "false." In other words, for the output to be 1, at least input [one] OR [two] must be 1. **For example, if a voter is permitted to vote either with [voter card] OR [aadhar card], the logic of OR GATE will be implied.**

Fig. 1.5 shows the pin diagram of IC 7432. It comprises of four OR gates, one V_{CC} and one Ground.

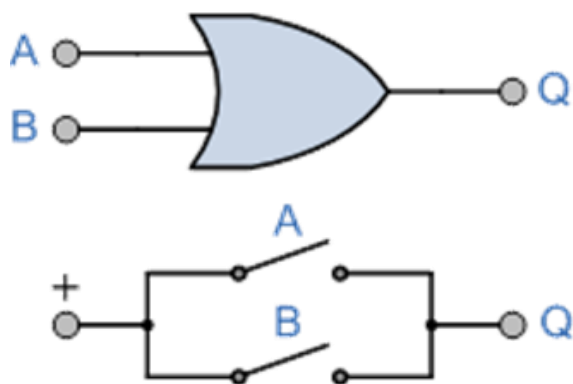


Figure 1.4

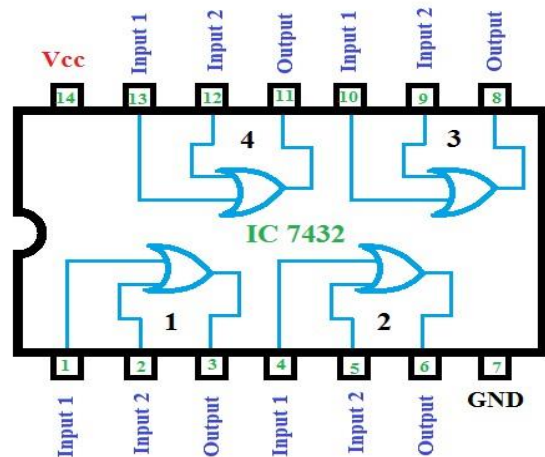


Figure 1.5

Table 1.2

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

The **XOR (exclusive-OR) gate**: Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same. **For example, if in a program a couple is the necessary condition to attend, the logic of XOR GATE will be implied.**

Fig. 1.7 shows the pin diagram of IC 7486. It comprises of four XOR gates, one V_{CC} and one Ground.

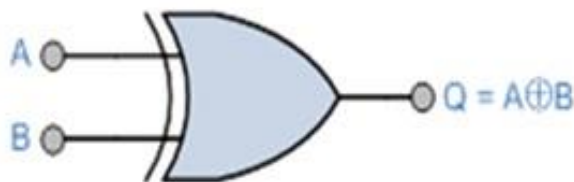


Figure 1.6

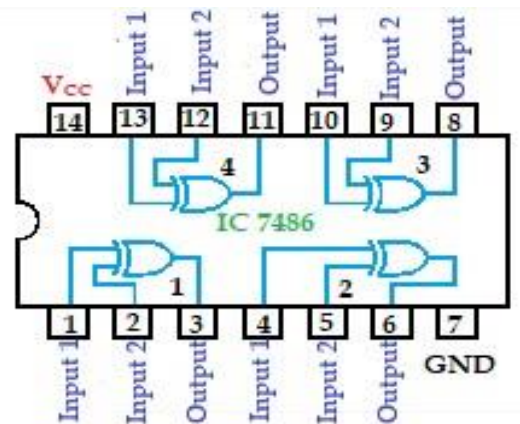


Figure 1.7

Table 1.3

A	B	Q
0	0	0
0	1	1
1	0	1
1	0	1

A **logical inverter**, sometimes called a *NOT gate* has only one input. It reverses the logic state. If the input is 1, then the output is 0. If the input is 0, then the output is 1.

Fig. 1.9 shows the pin diagram of IC 7404. It comprises of six NOT gates, one V_{CC} and one Ground.

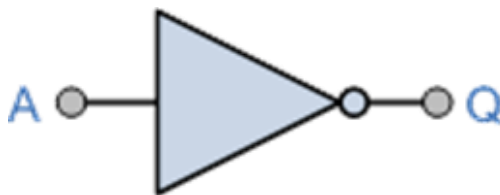


Figure 1.8

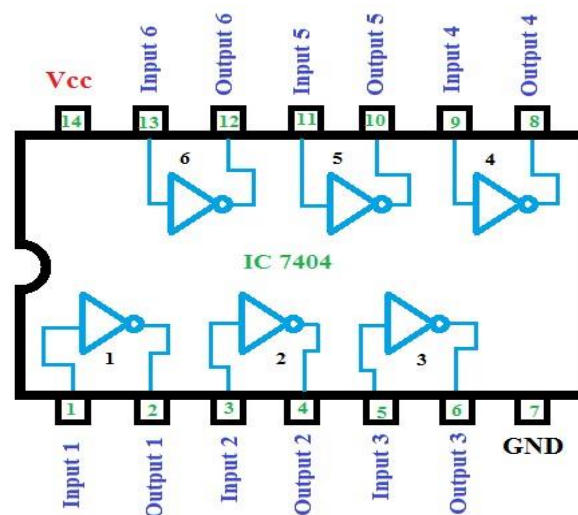


Figure 1.9

TABLE 1.4

A	Q
1	0
1	0

Experiment 1.1: a. To verify the truth tables of NAND and NOR gates.

b. To configure NAND and NOR gates as other logic gates and verify their truth tables.

Apparatus: Experimental board comprising of IC 7400 (quad NAND gates), IC 7402 (quad NOR gates), LED's, DMM, +5 V power supply, connecting wires.

Theory: Fig. 1.10 shows the symbol for NAND gate. The *NAND gate* operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "and" followed by negation. The output is "false" if both inputs are "true." Otherwise, the output is "true."

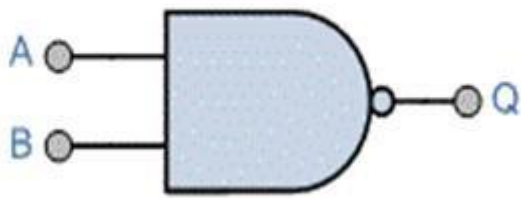


Figure 1.10

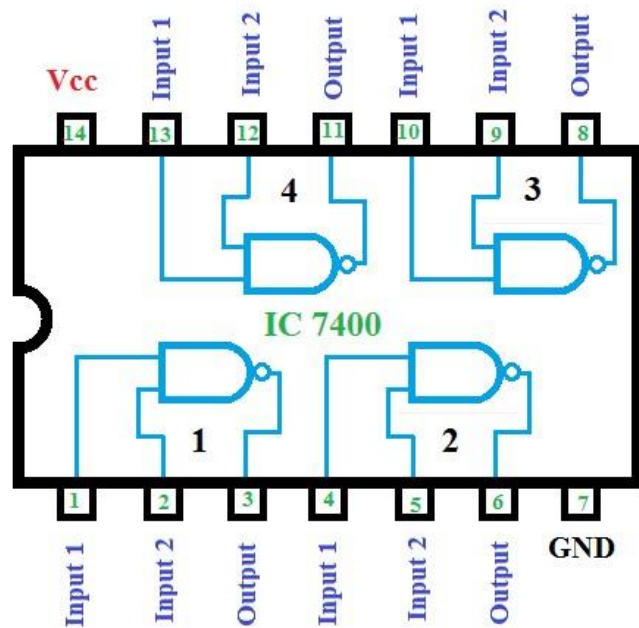


Figure 1.11

Fig. 1.11 shows the pin diagram of IC 7400. It comprises of four NAND gates, one V_{CC} and one Ground.

Table 1.5

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 1.12 shows the symbol for NOR gate. The *NOR gate* is a combination OR gate followed by an inverter. Its output is "true" if both inputs are "false." Otherwise, the output is "false."

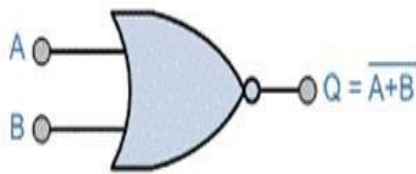


Figure 1.12

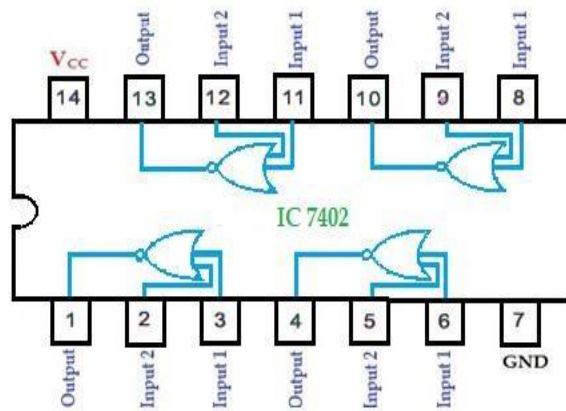


Figure 1.13

Fig. 1.13 shows the pin diagram of IC 7402. It comprises of four NOR gates, one V_{CC} and one Ground.

Table 1.6

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

A *universal gate* is a logic gate which can implement any Boolean function without the need to use any other type of logic gate. The NAND & NOR are universal gates. This means that any Boolean expression can be created using only NAND gates or only NOR gates.

Fig. 1.14 shows the configuration of NAND gate as NOT, AND & OR gates.

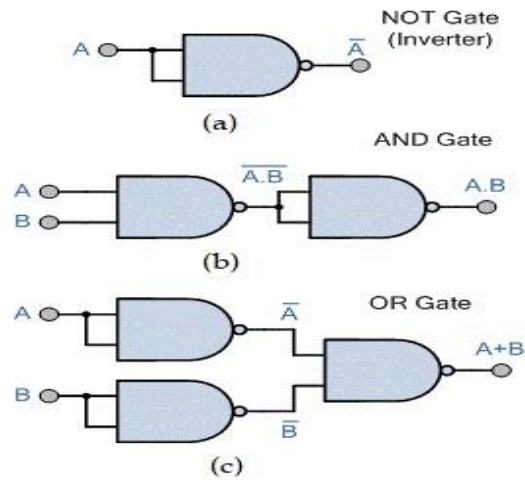


Figure 1.14

Fig. 1.15 shows the configuration of NOR gate as NOT, OR & AND gates.

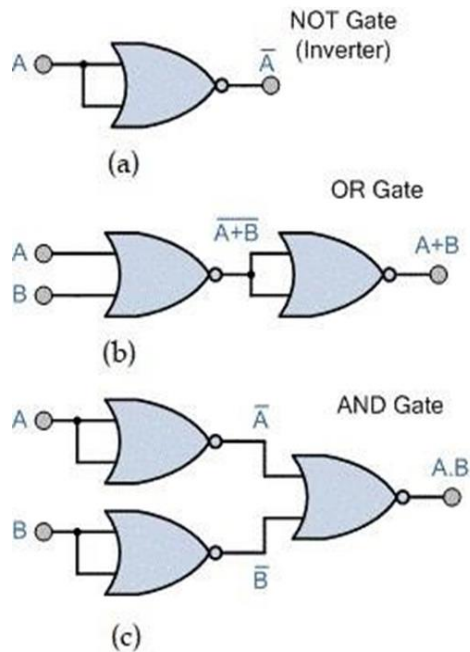


Figure 1.15

Procedure:

(A) NAND Gate:

- a. On the experimental board for NAND gate, make the appropriate connections for V_{CC} and G_{ND} .
- b. Verify the truth table for each NAND gate by checking the output at the LED.
- c. Configure the NAND gates for NOT, AND & OR gate as shown in Fig.

1.14 (a), (b) and (c) respectively and verify their truth tables.

(B) NOR Gate:

- a. On the experimental board for NOR gate, make the appropriate connections for V_{CC} and G_{ND} .
- b. Verify the truth table for each NOR gate by checking the output at the LED.
- c. Configure the NOR gates for NOT, OR & AND gate as shown in Fig.

1.15 (a), (b) and (c) respectively and verify their truth tables.

Observations:

(A) NAND Gate:

A	B	Y_1	Y_2	Y_3	Y_4
0	0				
0	1				
1	0				
1	1				

(A- 1) NAND Gate as NOT Gate:

A	Y
1	
0	

(A - 2) NAND Gate as AND Gate:

A	B	Y
0	0	
0	1	
1	0	
1	1	

(A - 3) NAND Gate as OR Gate:

A	B	Y
0	0	
0	1	
1	0	
1	1	

(B) NOR Gate:

A	B	Y ₁	Y ₂	Y ₃	Y ₄
0	0				
0	1				
1	0				
1	1				

(B- 1) NOR Gate as NOT Gate:

A	Y
1	
0	

(B - 2) NOR Gate as OR Gate:

A	B	Y
0	0	
0	1	
1	0	
1	1	

(B - 3) NOR Gate as AND Gate:

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

Result:

- a. The truth tables for NAND gate and NOR gate are verified successfully.
- b. NAND gates and NOR gates are successfully configured as other logic gates and their truth tables are also verified.

Precautions and Sources of Error:

- 1. The connections should be made according to the IC pin diagram.
- 2. The connections should be tight & V_{CC} should be applied at the specified pin only.

1.3 BOOLEAN ALGEBRA

Boolean algebra uses a set of laws and rules to define the operation of a digital logic circuit with 0's and 1's being used to represent a digital input or output condition.

Boolean algebra uses these zeroes and ones to create truth tables and mathematical expressions to define the digital operation of a logic *AND*, *OR* and *NOT* operations as well as ways of expressing other logical operations such as the *XOR* function.

1.4 De - MORGAN'S THEORY

De Morgan's theorems are basically two sets of rules or laws developed from the Boolean expressions for *AND*, *OR* and *NOT* using two input variables, *A* and *B*. These two rules or theorems allow the input variables to be negated and converted from one form of a Boolean function into an opposite form.

Experiment 1.2: To verify the De - Morgan's theorems.

Apparatus: Experimental board comprising of IC 7408 (quad *AND* gates), IC 7432 (quad *OR* gates), IC 7404 (*NOT* gates), LED's, DMM, +5 V power supply, connecting wires.

Theory: De - Morgan's first theorem states that two (or more) variables *NOR*'ed together is the same as the two variables inverted and *AND*'ed, while the second theorem states that two (or more) variables *NAND*'ed together is the same as the two variables inverted and *OR*'ed.

De - Morgan's First Theorem

De Morgan's first theorem proves that when two (or more) input variables are *AND*'ed and negated, they are equivalent to the *OR* of the complements of the individual variables. Thus, the equivalent of the *NAND* function will be a negative of the *OR* function, proving that $\overline{A \cdot B} = \overline{A} + \overline{B}$.

Fig. 1.16 shows the implementation of De Morgan's first theorem using logic gates.

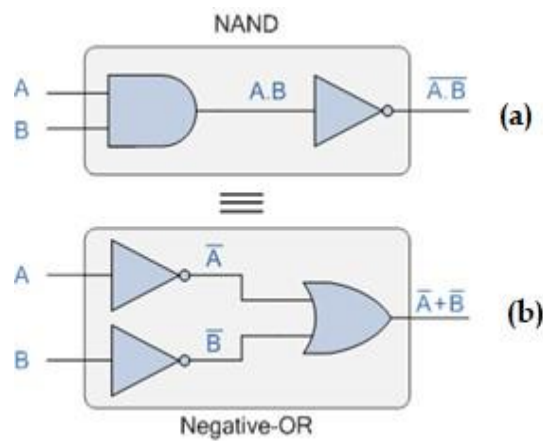


Figure 1.16

De - Morgan's Second Theorem

De Morgan's second theorem proves that when two (or more) input variables are OR'ed and negated, they are equivalent to the AND of the complements of the individual variables. Thus, the equivalent of the NOR function is a negative - AND function, proving that; $\overline{A+B} = \overline{A} \cdot \overline{B}$.

Fig. 1.17 shows the implementation of De Morgan's second theorem using logic gates.

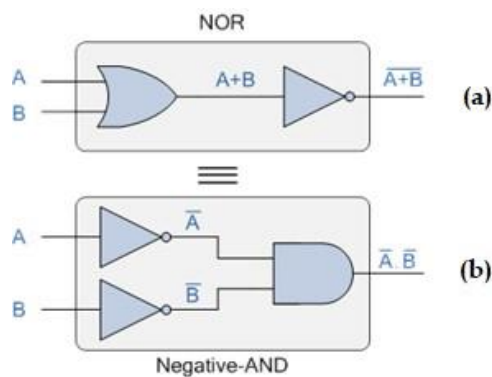


Figure 1.17

Procedure:

(A) AND, OR and NOT Gates:

- a. On the experimental board make the appropriate connections for V_{CC} & GND.
- b. Verify the truth table for each AND, OR and NOT gate by checking the output at the LED.

De – Morgan’s First Theorem:

- a. Configure the board for L.H.S of De – Morgan’s first theorem as shown in Fig. 1.16 (a) and verify the truth table for the same.
- b. Configure the board for R.H.S of De – Morgan’s first theorem as shown in Fig. 1.16 (b) and verify the truth table for the same.

De – Morgan’s Second Theorem:

- a. Configure the board for L.H.S of De – Morgan’s second theorem as shown in Fig. 1.17 (a) and verify the truth table for the same.
- b. Configure the board for R.H.S of De – Morgan’s second theorem as shown in Fig. 1.17 (b) and verify the truth table for the same.

Observations:

(A- 1) NOT Gate:

A	Y
1	
0	

(A - 2) AND Gate:

A	B	Y
0	0	
0	1	
1	0	
1	1	

(A - 3) OR Gate:

A	B	Y
0	0	
0	1	
1	0	
1	1	

(B - 1) De - Morgan's First Theorem (L.H.S):

A	B		
0	0		
0	1		
1	0		
1	1		
		A · B	A · B

(B-2) De - Morgan's First Theorem (R.H.S):

A	B		
0	0		
0	1		
1	0		
1	1		
		A	B
			A + B

(C - 1) De - Morgan's Second Theorem (L.H.S):

A	B	$A + B$	$\overline{A+B}$
0	0		
0	1		
1	0		
1	1		

(B - 2) De - Morgan's Second Theorem (R.H.S):

A	B	$A \cdot B$
0	0	
0	1	
1	0	
1	1	

Result: De - Morgan's theorems are verified successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.

1.4 BINARY ADDITION OF 2/3 - BITS

Binary adder is one of the basic combinational logic circuits. The outputs of a combinational logic circuit depend on the present input only. In other words, outputs of combinational logic circuit do not depend upon any previously applied inputs. Binary adder is one of the basic combinational logic circuits as present state of input variables.

Binary adders are basically classified into two types: Half Adder and Full Adder. The half adder circuit has two inputs: A and B, which add two input digits and generates a carry and a sum. The full adder circuit has three inputs: A, B and C, which adds three input numbers and generates a carry and sum.

Experiment 1.3: To study Half adder and Full Adder using Logic gates.

Apparatus: Experimental board comprising of IC 7408 (quad AND gates), IC 7432 (quad OR gates), IC 7486 (quad XOR gates), +5 V power supply, connecting wires, DMM.

Theory:

HALF ADDER

A half adder adds two binary digits where the input bits are termed as augend and addend and the result will be two outputs; one is the sum and the other is carry. Fig. 1.18 shows the block diagram of Half adder.



Figure 1.18

For addition of both the bits, XOR gate generates SUM while AND gate generates CARRY.

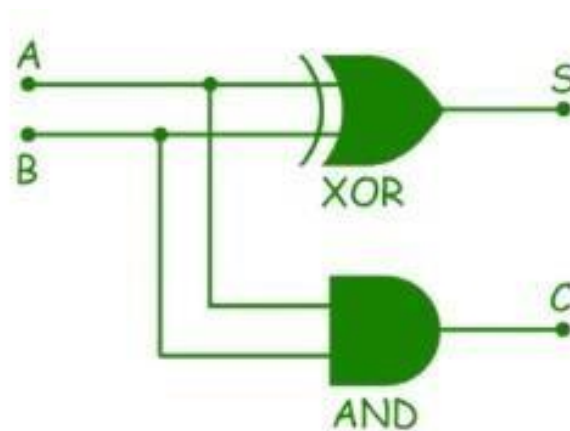


Figure 1.19

FULL ADDER

A full adder circuit adds three bits, where two of the three bits can be referred to as operands and the other is termed as bit carried in. The produced output is 2 - bit

output and can be referred to as output carry and sum. Fig. 1.20 shows the block diagram of Full adder.



Figure 1.20

A full adder circuit can be implemented with the help of two half adder circuits as shown in Fig. 1.21.

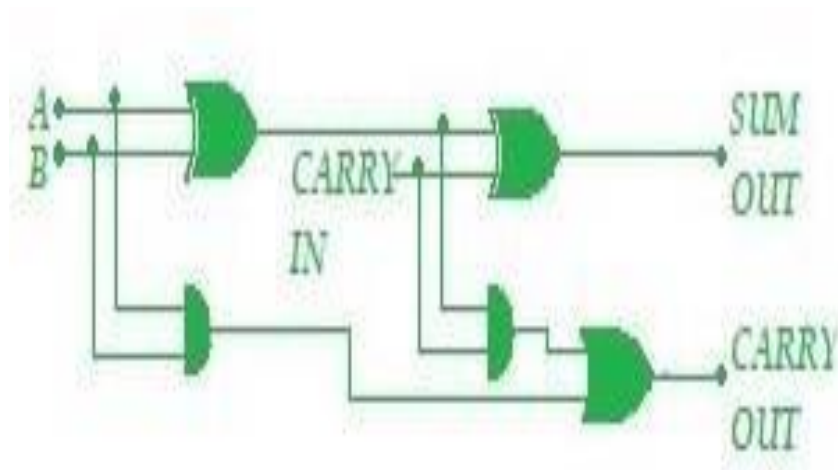


Figure 1.21

Initially, the half adder will be used to add A and B to produce a partial SUM and a second half adder logic can be used to add $C-IN$ to the sum produced by the first half adder to the final $SUM OUT$. If any of the half adder logic produces a

carry, there will be an output carry. So, *CARRY OUT* will be an *OR* function of the half adder Carry outputs.

Procedure:

(A) AND, OR and XOR Gates:

- a. On the experimental board make the appropriate connections for V_{CC} & GND.
- b. Verify the truth table for each AND, OR and XOR gate by checking the output in DMM.

(B) Half Adder:

- a. Configure the board for Half Adder as shown in Fig. 1.19 and verify the truth table for the same.
- b. Also measure the output voltage in DMM.

(C) Full Adder:

- a. Configure the board for Full Adder as shown in Fig. 1.21 and verify the truth table for the same.
- b. Also measure the output voltage in DMM.

Observations:

(A - 1) AND Gate:

A	B	Y	Output Voltage (V)
0	0		
0	1		
1	0		
1	1		

(A - 2) OR Gate:

A	B	Y	Output Voltage (V)
0	0		
0	1		
1	0		
1	1		

(A - 3) XOR Gate:

A	B	Y	Output Voltage (V)
0	0		
0	1		
1	0		
1	1		

(B) Half Adder

INPUTS		OUTPUT			
A	B	SUM		CARRY	
		Y	Output Voltage (V)	Y	Output Voltage (V)
0	0				
0	1				
1	0				
1	1				

(C) Full Adder

INPUTS			OUTPUT			
A	B	C	SUM		CARRY	
			Y	Output Voltage (V)	Y	Output Voltage (V)
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Result: The output of Half adder and Full adder circuits using Logic Gates are verified successfully. .

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.

1.5 BINARY ADDITION & SUBTRACTION OF 4-BIT NUMBERS

In digital circuits, a Binary Adder - Subtractor is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds. It is one of the components of the ALU.

Let us consider two 4 - bit binary numbers A and B as inputs. The circuit consists of 4 full adders since we are performing operation on 4 - bit numbers as shown in Fig. 1.22. There is a control line K that holds a binary value of either 0 or 1 which determines that the operation being carried out is addition or subtraction.

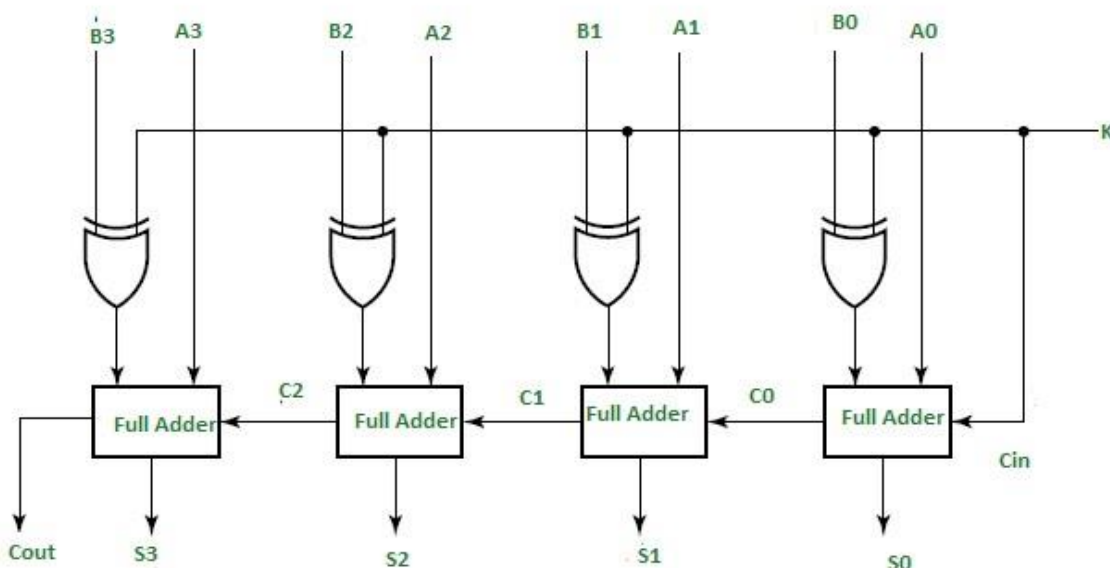


Figure 1.22

As shown above, the first full adder has control line directly as its input (input carry C_{in}), the input A_0 (the least significant bit of A) is directly input in the full adder.

The third input is the *XOR* of *B* and *K*. The two outputs produced are sum/difference (*S0*) and carry / borrow (*C0*).

If the value of *K* is 1, the operation being performed on the four bit numbers is subtraction. Similarly, if the value of *K* is 0, the operation is binary addition.

Then *C0* is serially passed to the second full adder as one of its inputs and so on. *C3* becomes the total carry to the sum/difference. *S1*, *S2*, *S3* are recorded to form the result with *S0*.

Experiment 1.4: To construct and study addition & subtraction using IC 7483 and IC 7486.

Apparatus: IC 7483, IC 7486, 5 V power supply, bread board, DMM, connecting wires.

Theory: Fig. 1.23 shows the pin diagram of IC 7483. It has a total of 16 pins. 4 *A*'s, 4 *B*'s, 4 *S*'s, one *C_{in}*, one *C_{out}*, one *V_{CC}* and one ground.

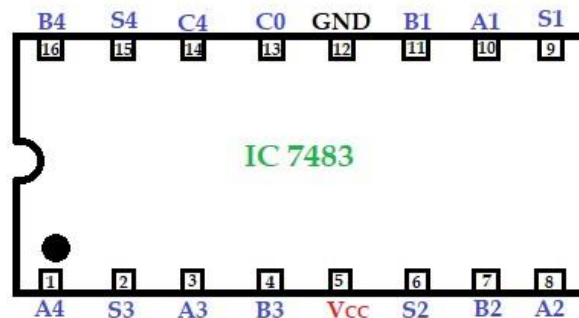


Figure 1.23

IC 7483 is an adder IC that can add two 4 – bit numbers. The largest number that can be obtained from a 4 – bit full adder is $(31)_{10}$ i.e., $(1F)_{16}$.

IC 7483 has 9 input pins. Out of these 4 input pins are for 1st data or addend (A), 4 input pins for 2nd data or augend (B) and the 9th input pin for the value of input carry (C0). IC 7483 has 5 output pins. Out of this 4 are for the sum (S) and last pin for output carry (C4).

IC 7483 can be made to act as a subtractor by using XOR gate as shown in Fig. 1.24.

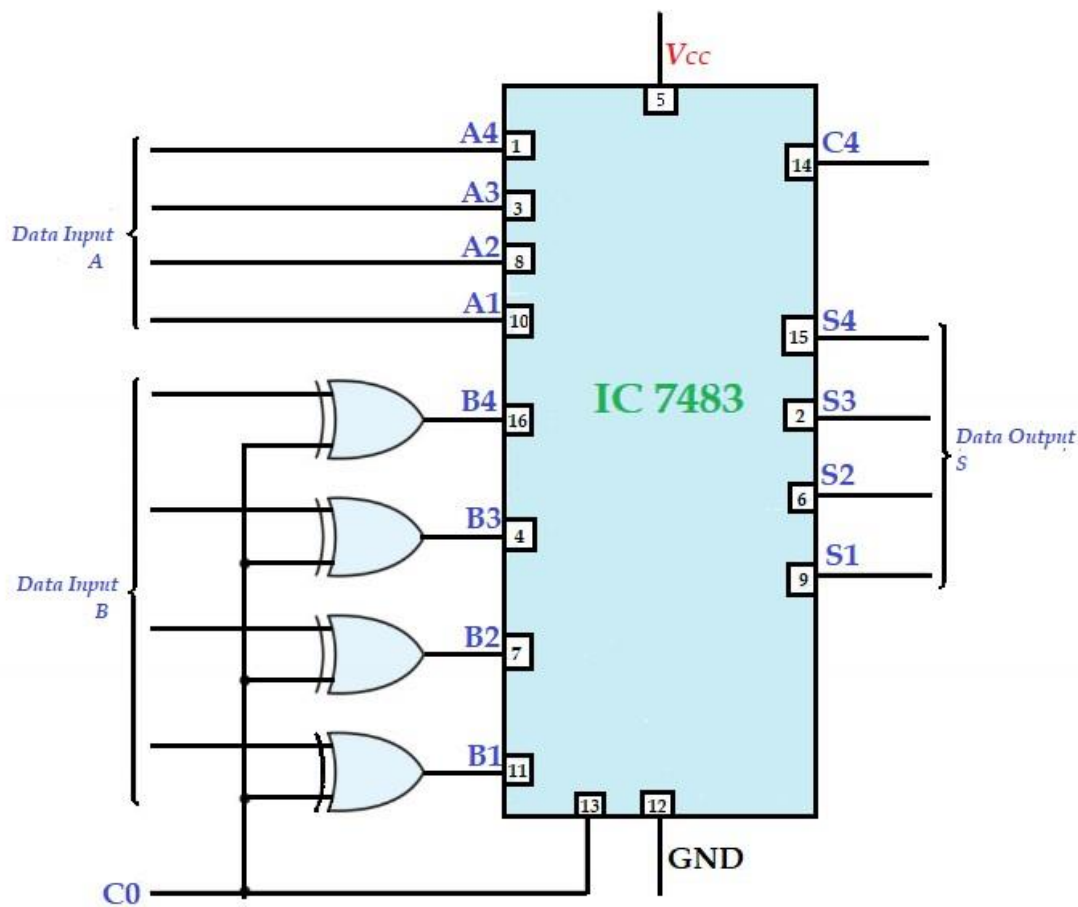


Figure 1.24

When $C0$ is 0, the IC acts as a full adder and when it is 1, the same configuration acts as a subtractor. When $C0$ is 0, one of the inputs of each EX-OR gate is 0 and the output of the gate is same as the value of other inputs, basically acts as buffer. Thus,

the gates give out the same output as the input and addition takes place. When C_0 is 1, one of the inputs of each EX-OR gate is 1 and the output of the gate is complement (1's complement) of the value of other inputs. Thus, the data reaching B_3, B_2, B_1 and B_0 is the 1's complement of the 2nd data and $C_0 = 1$, makes the data 2' complement. Hence, the 2's complement of the 2nd data is added to the 1st data, which is nothing but subtraction. Thus, the arrangement works as a subtractor.

An 8 - bit addition / subtraction can be formed by cascading two full adders as shown in Fig. 1.25.

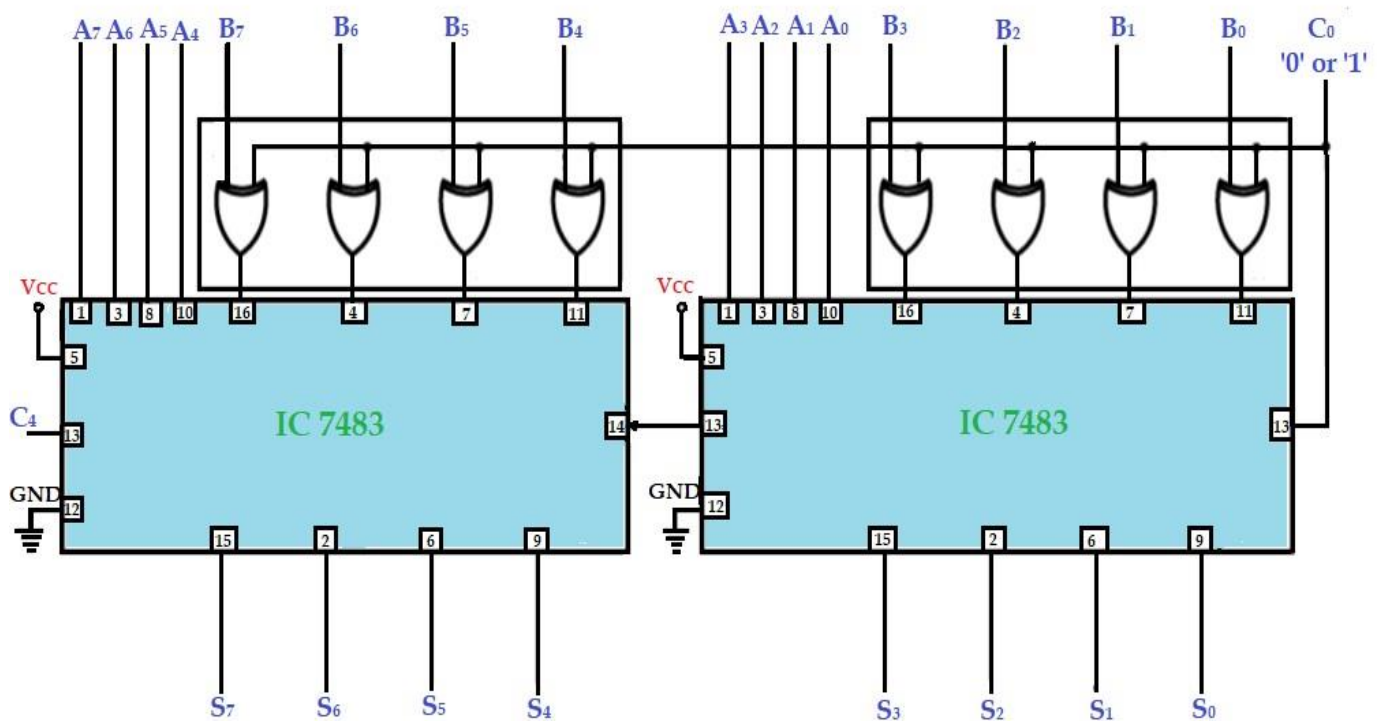


Figure 1.25

It can be directly connected to the CMOS, NMOS & TTL circuits. IC 7483 can operate with supply voltage of 4.75 to 5 V DC. It can operate properly at 0 to 70

degree Celsius temperature. IC 7483 is used in digital display driver circuits, counter circuits, calculator circuits and matrix keyboards, etc.

Procedure:

(A) 4 - Bit Addition/Subtraction

- a. Bread board the circuit as shown in Fig. 1.24 with making appropriate connections for V_{CC} and GND.
- b. For addition, make $C0$ as 0. Enter different combinations of data inputs of A's and B's and check the output in the DMM.
- c. For subtraction, make $C0$ as 1. Enter different combinations of data inputs of A's and B's and check the output in the DMM.

(B) 8 - Bit Addition/Subtraction

- a. Bread board the circuit as shown in Fig. 1.25 with making appropriate connections for V_{CC} and GND.
- b. For addition, make $C0$ as 0. Enter different combinations of data inputs of A's and B's and check the output in the DMM.
- c. For subtraction, make $C0$ as 1. Enter different combinations of data inputs of A's and B's and check the output in the DMM.

Observations:

(A) 4 - Bit Addition/Subtraction

1 st Data		2 nd Data		C ₀	Sum/Difference		C ₄
HEX	Binary	HEX	Binary		HEX	Binary	

(B) 8 - Bit Addition/Subtraction

1 st Data		2 nd Data		C ₀	Sum/Difference		C ₄
HEX	Binary	HEX	Binary		HEX	Binary	

Result: 4 - bit and 8 - bit adder/subtractor circuits using IC 7483 and IC 7486 have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.
3. O/Ps can be checked using LEDs connected with appropriate value of resistors or using DMM.

QUESTIONS FOR VIVA

Universal Gates

Q. Why the name universal associated with NAND & NOR gates?

Q. How will you configure universal gates as inverters & buffers?

De - Morgan's Theorem

Q. What does you conclude from De-Morgan's theorems?

Q. Name all the gates necessary to prove De-Morgan's theorems?

Half Adder & Full Adder

Q. What is the result of adding B_{16} to F_{16} in a 4-bit adder?

Q. What is the result of adding $2B_{16}$ to FF_{16} in an 8-bit adder?

Adder/Subtractor

Q. What is the result of $F_{16} - A_{16}$ in a 4-bit subtraction?

Q. What is the result of $7C_{16} - 6F_{16}$ in an 8-bit subtraction?

2

Digital Electronics

2.1 MULTIPLEXERS

Multiplex means *many into one*. A *multiplexer* is a circuit with many inputs but only one output. **By applying control signals, we can steer any input to the output, the control signal not only selects a particular input but also deselects all other inputs.** Fig. 2.1 illustrates a general multiplexer. The circuit has ' n ' input signals, ' m ' control signals and one output signal.

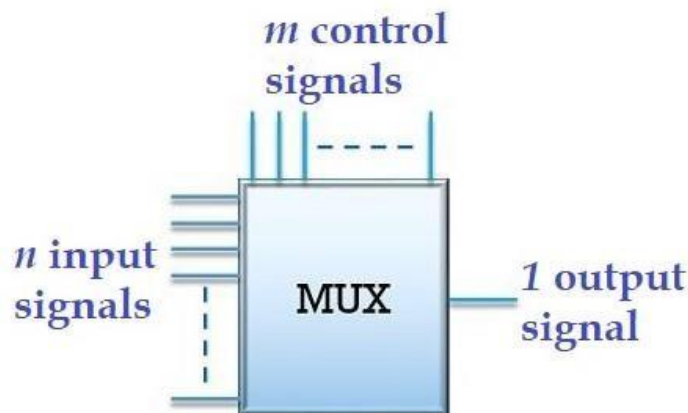


Figure 2.1

The select line determines which input is connected to the output and increase the amount of data that can be sent over a network within a certain time. It is also called a data selector. The multiplexer is used to perform high speed switching and is constructed by electronic components.

Applications of Multiplexer:

- Communication system
- Computer memory
- Telephone network
- Transmission from the computer system of a satellite

2.1.1 8:1 MULTIPLEXER

The 8:1 multiplexer comprises of 8 - input bits, one output bit and 3 - control bits. The eight inputs are namely $D_0 - D_7$; only one of the input bit is transmitted to the output. The output Y depends on the value of control inputs CBA . The control bits CBA decides which of the input data bit should transmit to the output. Fig. 2.2 shows the 8:1 multiplexer circuit diagram using AND gates.

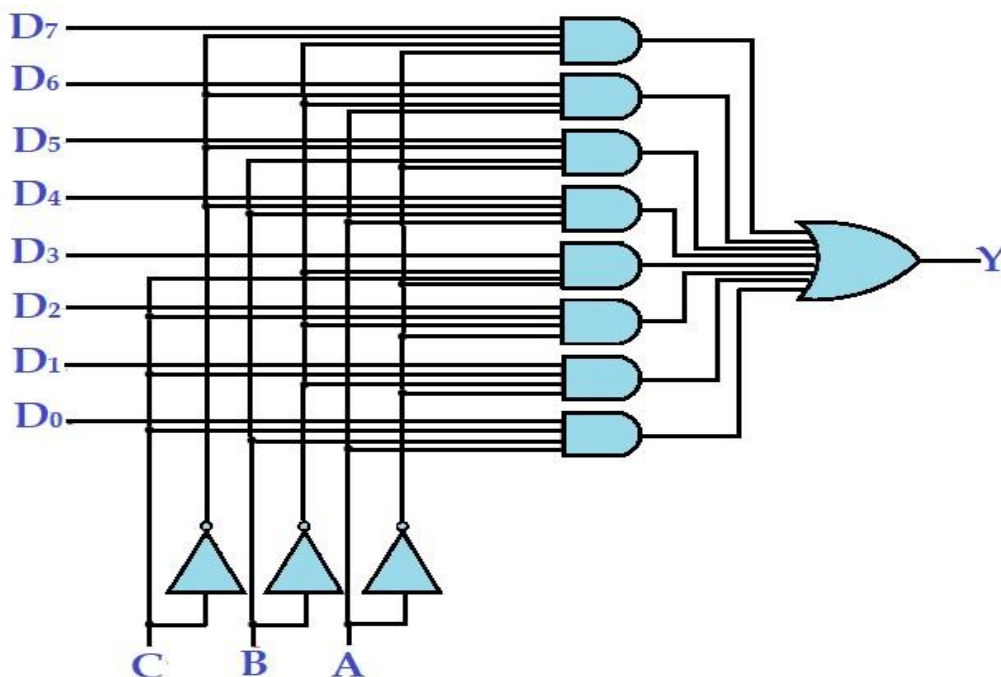


Figure 2.2

For example, when the control bits $CBA = 000$, then the bottom AND gate is allowed while remaining AND gates are restricted. Thus, data input D_0 is transmitted to the output Y . If the control bits changes to $CBA = 111$, all gates are disabled except the highest AND gate. In this case, D_7 is transmitted to the output Y .

Experiment 2.1.1: To study the working of IC 74151 (8:1 Multiplexer).

Apparatus: IC 74151, 5 V power supply, bread board, DMM, connecting wires.

Theory: IC 74151 is an 8:1 multiplexer. It is a 16 pin IC comprising of 8 inputs ($D_0 - D_7$), two outputs (Y & Y'), three control bits (C, B, A), one ENABLE pin E , one V_{CC} and one ground as shown in Fig. 2.3.

The two outputs are complementary, provides an open option for further cascading. **An ENABLE input is provided, which when at high level disables all data inputs and forces Y output to low state and Y' output to high state. Implies that, even after energizing the IC with V_{CC} , the IC is ready to perform only after ENABLE pin is made zero.**

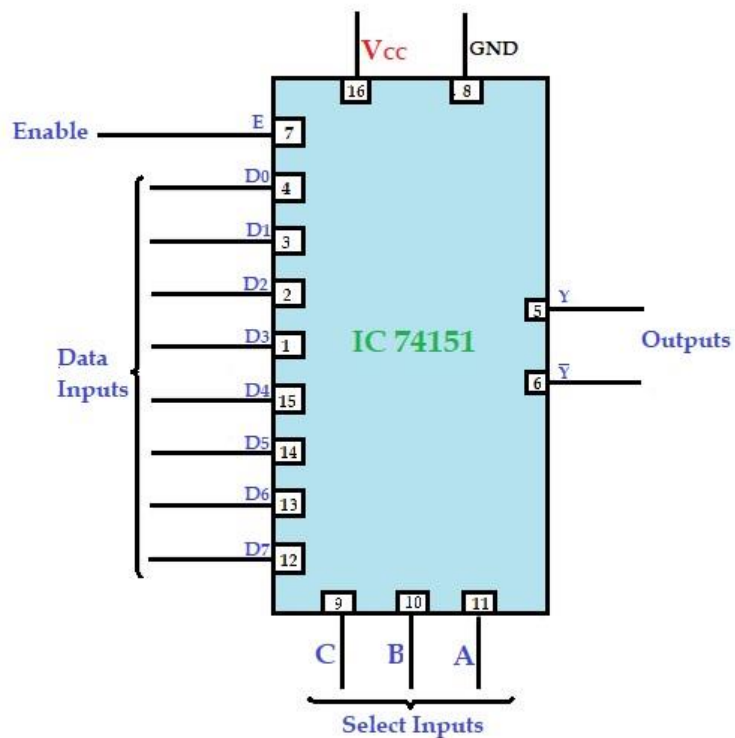


Figure 2.3

Procedure:

- Bread board the circuit as shown in Fig. 2.3 with making appropriate connections for V_{CC} and GND.
- Keeping Enable pin E as HIGH, select inputs (C, B, A) as X (either 0 or 1), note the corresponding outputs at (Y & Y').
- Keep E as LOW, vary the select inputs (C, B, A) and note the outputs for the corresponding input.

Observations:

INPUTS				OUTPUTS	
E	C	B	A	INPUT	Y
H	X	X	X	X	
L	0	0	0	D0 =	
L	0	0	1	D1 =	
L	0	1	0	D2 =	
L	0	1	1	D3 =	
L	1	0	0	D4 =	
L	1	0	1	D5 =	
L	1	1	0	D6 =	
L	1	1	1	D7 =	

Result: The working of 8:1 Multiplexer IC 74151 with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight, loose connection can be misunderstood as low rather than high.
3. Same power supply the one used to energize the IC can be used appropriately for inputs & control pins.
4. V_{CC} and ground should be applied carefully at the specified pins only.

5. Based on the IC type sometime enable pin is active low & sometime active high.
6. The O/P can be verified either using LED preceded by a resistor or on DMM.
7. Until the IC is used as an interfacing component in a bigger circuit, the MUX action can be verified using few inputs statistically.

Experiment 2.1.2: To construct a 16:1 multiplexer from two 8: 1 MUX (IC 74151) and also study the working of IC 74150 (16: 1 Multiplexer).

Apparatus: IC 74151, IC 74150, IC 7402, 5 V power supply, bread board, DMM, connecting wires.

Theory: A 16: 1 multiplexer can be constructed using two 8:1 multiplexers (IC 74151) and one IC 7402 consisting of NOR gate to be configures as inverter as shown in Fig. 2.4.

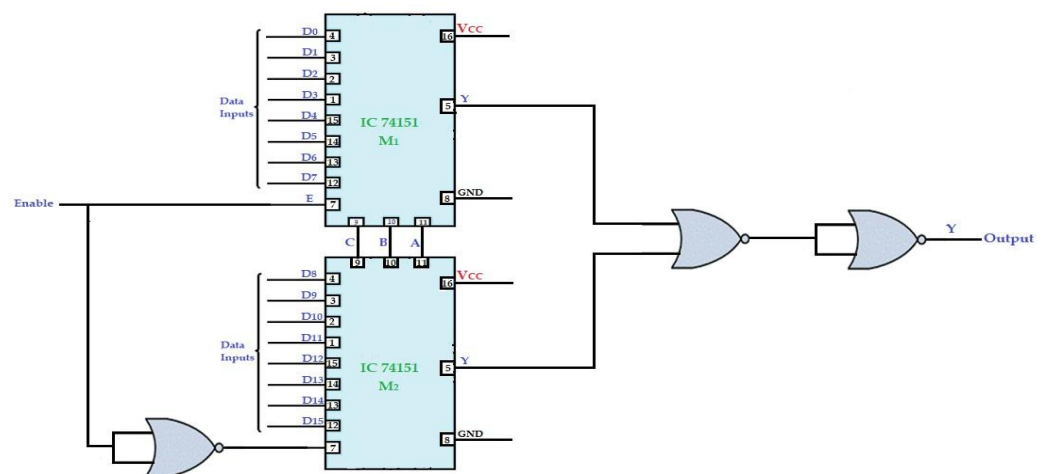


Figure 2.4

It consists of 4 select lines, 16 inputs and one output. The Enable E of the multiplexer M₁ is complimented and is connected to the enable of the second

multiplexer M_2 . The fourth select line D is \bar{E} , i.e., enable of the first multiplexer M_1 .

The select inputs of both the multiplexers are shorted. When the enable is LOW, multiplexer M_1 is active and M_2 remains off. When enable is HIGH, multiplexer M_1 is off and M_2 remains active.

IC 74150 selects one of sixteen data inputs. It has a strobe input which must be at a LOW logic level to enable the device. A HIGH level at the strobe forces the output HIGH. The IC 74150 features an inverted output only. Fig. 2.5 shows the pin diagram of IC 74150.

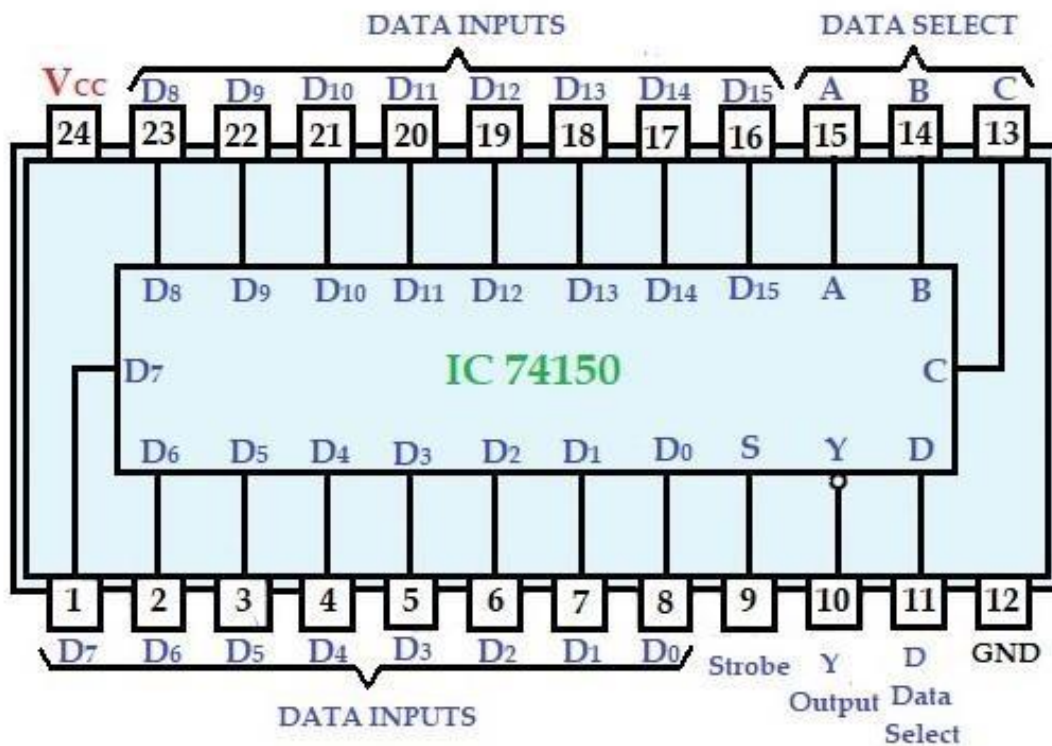


Figure 2.5

Procedure:

(A) 16:1 MUX from two 8:1 MUX using IC 74151:

- a. Bread board the circuit as shown in Fig. 2.4 with making appropriate connections for V_{CC} and GND.
- b. Keeping Enable pin E as LOW, vary the select inputs (C, B, A), inputs either 0 or 1 from D0 to D7 and note the corresponding outputs at Y.
- c. Keeping Enable pin E as HIGH, vary the select inputs (C, B, A), inputs either 0 or 1 from D8 to D15 and note the corresponding outputs at Y.

(B) 16:1 MUX using IC 74150:

- a. Bread board the circuit as shown in Fig. 2.5 with making appropriate connections for V_{CC} and GND.
- b. Keeping Enable pin E as LOW, vary the select inputs (C, B, A), inputs either 0 or 1 from D0 to D7 and note the corresponding outputs at Y.
- c. Keeping Enable pin E as HIGH, vary the select inputs (C, B, A), inputs either 0 or 1 from D8 to D15 and note the corresponding outputs at Y.

Observations:

16:1 MUX from two 8: 1 MUX using IC 74151:

E	C	B	A	INPUTS	OUTPUT
0	0	0	0	D0 =	
0	0	0	1	D1 =	
0	0	1	0	D2 =	
0	0	1	1	D3 =	
0	1	0	0	D4 =	
0	1	0	1	D5 =	
0	1	1	0	D6 =	
0	1	1	1	D7 =	
1	0	0	0	D8 =	
1	0	0	1	D9 =	
1	0	1	0	D10 =	
1	0	1	1	D11 =	
1	1	0	0	D12 =	
1	1	0	1	D13 =	
1	1	1	0	D14 =	
1	1	1	1	D15 =	

(A) 16:1 MUX using IC 74150:

E	C	B	A	INPUTS	OUTPUT
0	0	0	0	D0 =	
0	0	0	1	D1 =	
0	0	1	0	D2 =	
0	0	1	1	D3 =	
0	1	0	0	D4 =	
0	1	0	1	D5 =	
0	1	1	0	D6 =	
0	1	1	1	D7 =	
1	0	0	0	D8 =	
1	0	0	1	D9 =	
1	0	1	0	D10 =	
1	0	1	1	D11 =	
1	1	0	0	D12 =	
1	1	0	1	D13 =	
1	1	1	0	D14 =	
1	1	1	1	D15 =	

Result: A 16:1 multiplexer from two 8:1 MUX (IC 74151) is constructed and also the working of IC 74150 (16:1 Multiplexer) with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.
3. Same power supply the one used to energize the IC can be used appropriately for inputs & control pins.
4. Based on the IC type sometime enable pin is active low & sometime active high.
5. The O/P can be verified either using LED preceded by a resistor or on DMM.
6. Until the IC is used as an interfacing component in a bigger circuit, the MUX action can be verified using few inputs statistically.

2.2 DEMULTIPLEXERS

Demultiplex means *one into many*. A *demultiplexer* is a logic circuit with one input and many outputs. By applying control signals, we can steer the input signal to one of the output lines. Fig. 2.6 illustrates a general demultiplexer. The circuit has one input signal, ' m ' control signals and ' n ' output signals.

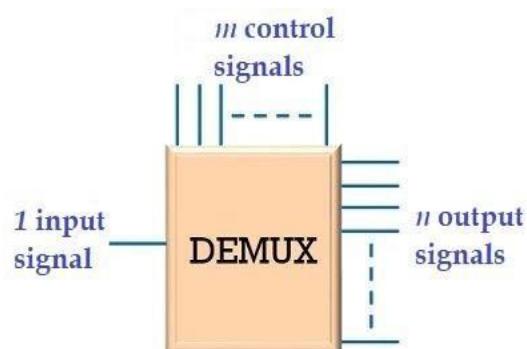


Figure 2.6

A demultiplexer is used to send a signal to one of the many devices, a demultiplexer does reverse to what the multiplexer does.

Applications of Demultiplexer:

- Communication system
- Arithmetic logic unit
- Serial to Parallel converter

2.2.1 1:4 DEMULTIPLEXER

The 1:4 de-multiplexer comprises of one input bit, 4 output bits and control bits (B, A). The input bit is considered as data D. This data bit is transmitted to the data bit of the output lines, which depends on the BA value. Fig. 2.7 shows the 1:4 de-multiplexer circuit diagram using AND gates.

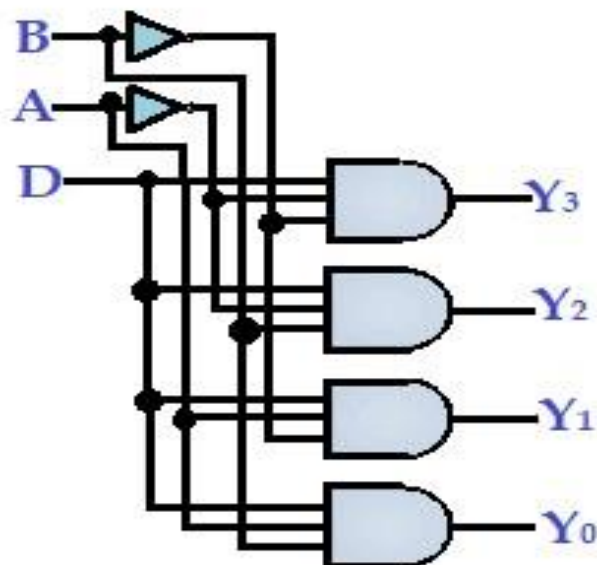


Figure 2.7

When the control input $BA = 01$, the second lowest AND gate is permitted while the remaining AND gates are restricted. Thus, only data bit D is transmitted to the output and $Y1 = \text{Data}$. If the data bit D is low, the output $Y1$ is low. If data bit D is high, the output $Y1$ is high. The value of the output $Y1$ depends upon the value of data bit D , the remaining outputs are in a low state.

If the control input changes to $AB = 10$, then all the gates are restricted except the second AND gate from the top. Then, data bit D is transmitted only to the output $Y2$ and $Y2 = \text{Data}$.

2.2.1 1: 8 DEMULTIPLEXER

The 1:8 demultiplexer is also called a data distributor as it requires one input, 3 selected lines and 8 outputs. It takes one single input data line and then switches it to any one of the output lines as shown in Fig. 2.8.

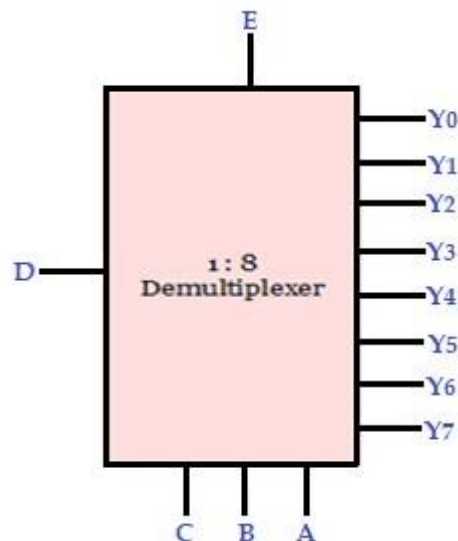


Figure 2.8

The working of 1: 8 demultiplex can be understood on the similar ground as 1:4 De - Mux.

Experiment 2.2: To study the working of IC 74155 (1: 4/8 Demultiplexer).

Apparatus: IC 74155, 5 V power supply, bread board, DMM, connecting wires.

Theory: 74155 is a 16 pin IC comprising of two select inputs (B and A), two data pins (C1 and C2), two STROBE pins (G1 and G2), eight outputs (1Y0 - 1Y3, 2Y0 - 2Y3), one V_{CC} and one ground.

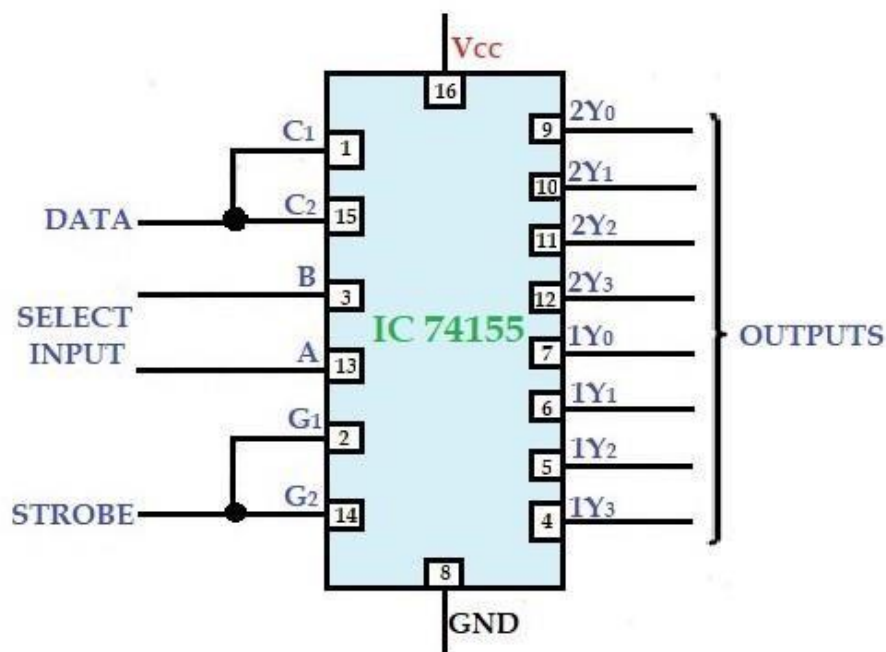


Figure 2.9

IC 74155 is a dual 1: 4 line demultiplexer with individual strobes and common binary address inputs in a single 16 - pin package. When both the sections are enabled by the strobes, the common address inputs sequentially select and route associated input data to the appropriate output of each section. The

individual strobes permit activating or inhibiting each of the 4 - bit sections as desired. Data applied to input C1 is inverted at its outputs and data applied at C2 is not inverted through its outputs. The inverter following the C1 data input permits use as a 3 - to - 8 line decoder or 1: 8 demultiplexer, without external gating.

Procedure:

(A) 1 : 4 Demultiplexer:

- a. Bread board the circuit as shown in Fig. 2.9 with making appropriate connections for V_{CC} and GND.
- b. Vary the select inputs A and B and observe that the input data is routed to the selected output (1Y0 - 1Y3) by keeping G1 as LOW. Note the corresponding output.
- c. Vary the select inputs A and B and observe that the input data is routed to the selected output (2Y0 - 2Y3) by keeping G2 as LOW. Note the corresponding output.

(B) 1 : 8 Demultiplexer:

- a. Bread board the circuit as shown in Fig. 2.9 with making appropriate connections for V_{CC} and GND.
- b. Connect C1 and C2 together to form select line C.
- c. Also connect G1 and G2 together to form strobe/data G.
- d. Vary the select inputs A, B, C and observe the corresponding outputs by keeping G as LOW.

Observations:

(A - 1) 1: 4 Demultiplexer:

INPUTS		OUTPUTS					
SELECT		STROBE	DATA	1Y3	1Y2	1Y1	1Y0
B	A	G1	C1				
X	X	H	X				
0	0	L	H				
0	1	L	H				
1	0	L	H				
1	1	L	H				
X	X	X	L				

(A - 2) 1: 4 Demultiplexer:

INPUTS		OUTPUTS					
SELECT		STROBE	DATA	2Y3	2Y2	2Y1	2Y0
B	A	G2	C2				
X	X	H	X				
0	0	L	L				
0	1	L	L				
1	0	L	L				
1	1	L	L				
X	X	X	H				

(B)1:8Demultiplexer

INPUTS			OUTPUTS								
SELECT			DATA/STROBE								
C	B	A	G	2Y3	2Y2	2Y1	2Y0	1Y3	1Y2	1Y1	1Y0
X	X	X	H								
0	0	0	L								
0	0	1	L								
0	1	0	L								
0	1	1	L								
1	0	0	L								
1	0	1	L								
1	1	0	L								
1	1	1	L								

Result: The working of 1: 4/8 de-multiplexer IC 74155 with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. Same power supply the one used to energize the IC can be used appropriately for inputs & control pins. V_{CC} and ground should be applied carefully at the specified pins only.
3. Based on the IC type sometime enable pin is active low & sometime active high.
4. The O/P can be verified either using LED preceded by a resistor or on DMM.

5. Until the IC is used as an interfacing component in a bigger circuit, the de-MUX action can be verified using few inputs statistically.

2.3 ENCODER

An encoder is a combinational circuit that encodes information from maximum of 2^n input lines into an n output lines. It will produce a binary code equivalent to the input, which is active HIGH.

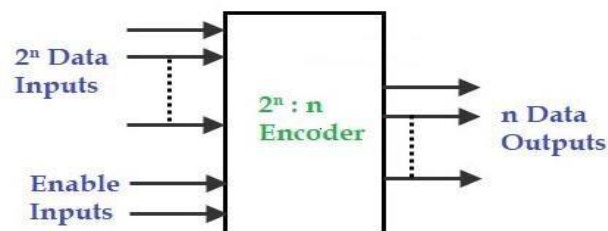


Figure 2.10

2.3.1 8:3 ENCODER (OCTAL TO BINARY)

The 8 to 3 Encoder or octal to Binary encoder consists of 8 inputs: Y_7 to Y_0 and 3 outputs: A_2 , A_1 & A_0 as shown in Fig. 2.11. Each input line corresponds to each octal digit and three outputs generate corresponding binary code.

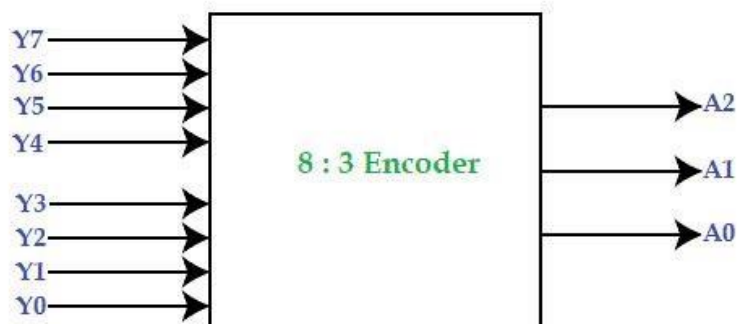


Figure 2.11

2.3.2 8:3 BIT PRIORITY ENCODER

An 8 to 3 priority encoder has 8 inputs: Y_7 to Y_0 and 3 outputs: A_2 , A_1 & A_0 . Here, the input Y_7 has the highest priority, whereas, the input, Y_0 has the lowest priority. For example, if input lines Y_2 , Y_3 and Y_5 are applied simultaneously, the output code would be for input Y_5 as this has the highest priority out of the three inputs. Once input Y_5 has been removed the next highest output code would be for input Y_3 and so on.

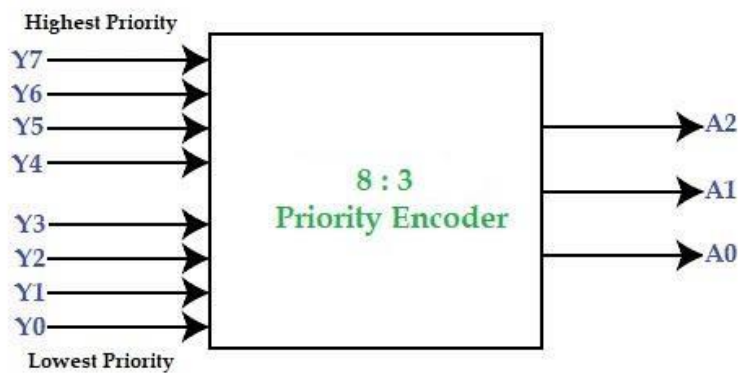


Figure 2.12

Experiment 2.3: To study the working of IC 74LS148 (8: 3 Priority Encoder).

Apparatus: IC 74LS148, 5 V power supply, bread board, DMM, connecting wires.

Theory: IC 74LS148 contain an 8 to 3 - bit priority encoder that converts 8 - data lines to 3 - line binary. The 74LS148 ICs have an operating range of V_{CC} from 4.5 V to 5.5 V. Maximum current consumption is 20 mA through I_{CC} and 0.2 mA per input pin. The maximum propagation delay is 15 ns. They can provide 0.4 mA at high input and 8 mA at low input. They allow inputs and outputs up to

V_{CC} . The recommended operating V_{CC} is 5 V. It's mostly used for Bit Encoding, Code Converters and Generators.

Fig. 2.13 shows the pin diagram of IC 74LS148. It comprises of 8 inputs, 3 outputs, one ENABLE INPUT (EI), one ENABLE OUTPUT (EO), one low GROUND SIGNAL (GS), one V_{CC} and one Ground.

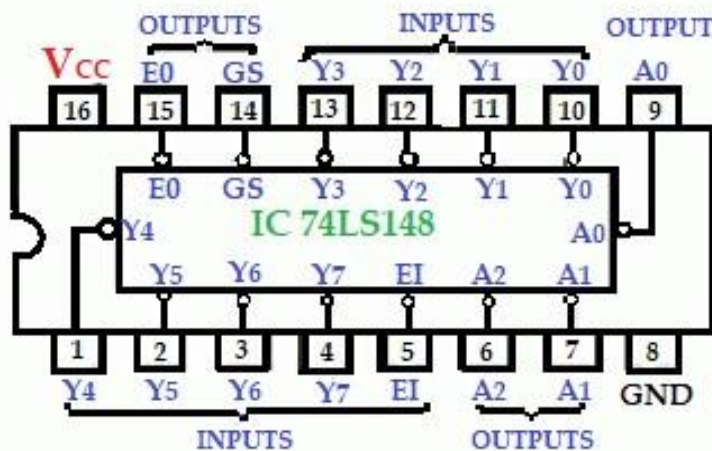


Figure 2.13

IC 74148 accepts data from eight active LOW inputs ($Y7 - Y0$) and provides a binary representation on the three active LOW outputs ($A2, A1, A0$). A HIGH on the ENABLE INPUT (EI) will force all outputs to the inactive (HIGH) state. GROUP SIGNAL (GS) is active LOW when any input is LOW; which indicates that any one of the inputs is active. ENABLE OUTPUT (EO) is active LOW when all inputs are HIGH. Both EO and GS are in the inactive HIGH state when the ENABLE INPUT (EI) is HIGH.

Procedure:

- a. Bread board the circuit as shown in Fig. 2.13 with making appropriate connections for V_{CC} and GND.
- b. Keeping EI at HIGH and $Y7 - Y0$ at X (either 0 or 1), note the outputs at A2, A1, A0, GS and EO.
- c. Keeping EI at LOW and $Y7 - Y0$ at HIGH, note the outputs at A2, A1, A0, GS and EO.
- d. Keeping EI at LOW and $Y7$ at LOW, note the outputs at A2, A1, A0, GS and EO.
- e. Similarly, go on making each input from $Y6 - Y0$ LOW one by one and note the corresponding outputs at A2, A1, A0, GS and EO.
- f. Now make any two inputs LOW and note the outputs.

Observations:

INPUTS									OUTPUTS				
EI	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	X	X	X	X	X	X	X	L	L	L	L	H
L	H	L	X	X	X	X	X	X	L	L	H	L	H
L	H	H	L	X	X	X	X	X	L	H	L	L	H
L	H	H	H	L	X	X	X	X	L	H	H	L	H
L	H	H	H	H	L	X	X	X	H	L	L	L	H

L	H	H	H	H	H	L	X	X	H	L	H	L	H
L	H	H	H	H	H	H	L	X	H	H	L	L	H
L	H	H	H	H	H	H	H	L	H	H	H	L	H
L	X	X	L	H	L	H	H	L	L	H	L	L	H
L	L	X	H	L	H	H	L	H	L	L	L	L	H

Result: The working of 8:3 Priority Encoder IC 74LS148 with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight. V_{CC} and ground should be applied carefully at the specified pins only.
3. Based on the IC type sometime enable pin is active low & sometime active high.
4. The O/P can be verified either using LED preceded by a resistor or on DMM.

2.4 DECODER

Decoder is a combinational circuit that has ' n ' input lines and maximum of 2^n output lines. One of these outputs will be active HIGH based on the combination of inputs present, when the decoder is enabled which implies that decoder detects a particular code.

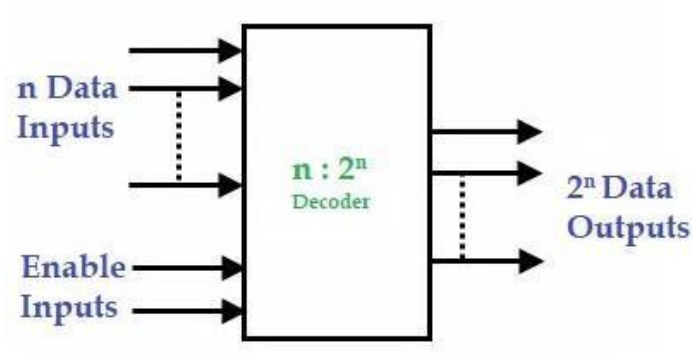


Figure 2.14

2.4.13: 8 DECODER (BINARY TO OCTAL)

The 3 to 8 line decoder is also known as Binary to Octal Decoder. In a 3 to 8 line decoder, there is a total of eight outputs, i.e., Y_0 to Y_7 and three inputs i.e., A_2 , A_1 , A_0 . This circuit has an enable input E.

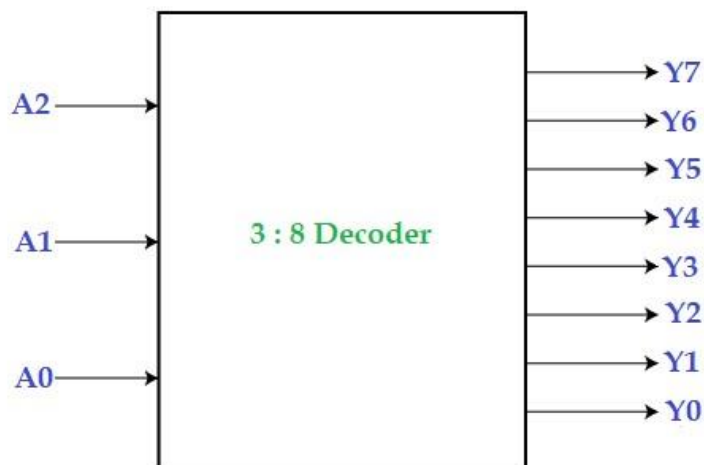


Figure 2.15

Experiment 2.4: To study the working of IC 74LS138 (3: 8 Decoder).

Apparatus: IC 74LS138, 5 V power supply, bread board, DMM, connecting wires.

Theory: IC 74LS138 is a TTL based logical device used to convert 3 bit binary data to 8 – bits. The actual purpose of this chip is designed for de – multiplexing or in machine language we can say as a decoding device. The device takes 3 input data and converts it to 8 – bit data. This chip includes three enable pins, three input pins and 8 output pins. The three enable pins come up as control pins that minimize the use of external gate and inverter when data is expanding.

Features:

- IC 74LS138 has internal ESD protection.
- The power input range for the decoder is 1.0 – 5.0 Volts.
- The typical propagation time for IC is 21 ns.
- Its low power consumption is almost 32 mW.
- The decoder can work in the temperature range from – 40 to + 125 degree.

Fig. 2.16 shows the pin diagram of IC 74LS138. It comprises of 3 inputs (A B, C), 8 outputs (Y7 – Y0), three ENABLE INPUTS (G1, G2A, G2B), one V_{CC} and one Ground.

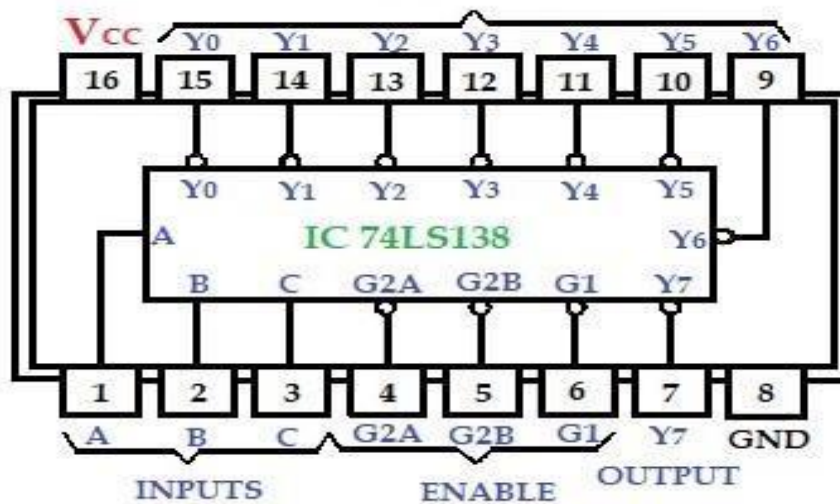


Figure 2.16

The decoder accepts three binary weighted inputs and when enabled provides eight mutually exclusive active low outputs. All NAND gates on output are controlled by enabling pins directly. Let the output only when two enable pins G2A and G2B is LOW and G1 is HIGH. If all the enable pins are active then there will be output, it doesn't matter input is HIGH or LOW. In case of any single enable pin is not active then there won't be any output, it doesn't matter what the inputs and other enable pins are. Every single output has a fixed input state. Every input makes the one output state LOW when enable pins are active.

Procedure:

- Bread board the circuit as shown in Fig. 2.16 with making appropriate connections for V_{CC} and GND.
- Keeping G1 and G2B at X (either 0 or 1), and G2A at HIGH, note the corresponding outputs for inputs C, B, A all kept at X.

- c. Making G1 as LOW and the rest of the inputs as X, note the corresponding outputs.
- d. Now keeping G1 as HIGH and the rest of the inputs LOW, note the corresponding outputs.
- e. Keep varying the three inputs C, B, A; note the corresponding outputs.

Observations:

INPUTS						OUTPUTS							
G1	G2A	G2B	C	B	A	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H	L
H	L	L	L	L	H	H	H	H	H	H	H	L	H
H	L	L	L	H	L	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	L	H	H	H
H	L	L	H	L	L	H	H	H	L	H	H	H	H
H	L	L	H	H	L	H	L	H	H	H	H	H	H
H	L	L	H	H	H	L	H	H	H	H	H	H	H

Result: The working of 3: 8 Decoder IC 74LS138 with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.

2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.

2.5 SEVEN SEGMENT DISPLAY

A *7-segment display* is used as numerical indicator and consists of a number of LEDs arranged in seven segments identified as *a, b, c, d, e, f, g* as shown in Fig. 2.17. Any number between 0 and 9 can be indicated by lighting the appropriate segments. An additional LED is used for the indication of a decimal point (DP).

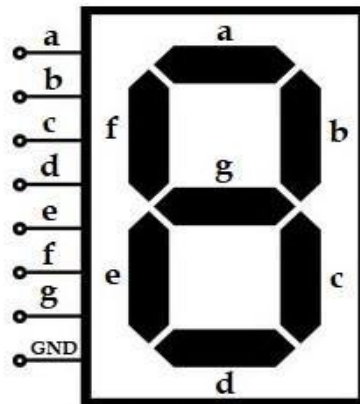


Figure 2.17

The 7-segment displays are of two types: Common Anode & Common Cathode type.

2.5.1 COMMON CATHODE TYPE

In the common cathode display, all the cathode connections of the LED segments are joined together to logic 0 or ground. The individual segments are

illuminated by application of a HIGH or logic 1 signal via a current limiting resistor to forward bias the individual LEDs ($a - g$).

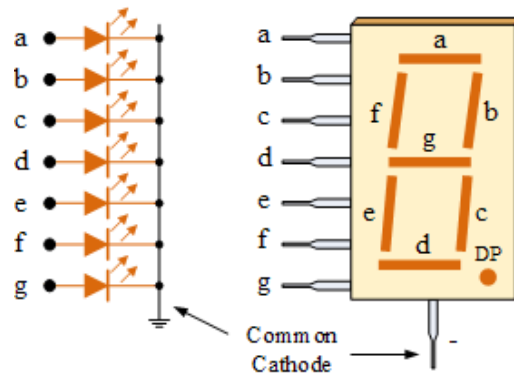


Figure 2.18

2.5.2 COMMON ANODE TYPE

In the common anode display, all the anode connections of the LED segments are joined together to logic 1. The individual segments are illuminated by applying a ground, logic 0 or LOW signal via a suitable current limiting resistor to the cathode of the particular segment ($a - g$).

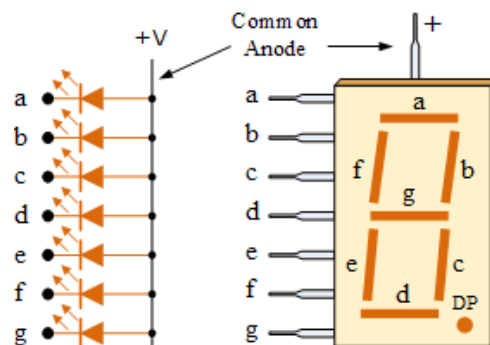


Figure 2.19

Experiment 2.5: To study the working of seven segment display IC 74LS47/74LS48.

Apparatus: Seven Segment Display (FND500/FND507), 5 V power supply, connecting wires, decoder/driver IC 74LS47/74LS48.

Theory: The IC 74LS47 has four inputs for the BCD digits A, B, C & D and outputs for each of the segments of the 7-segment display. Note that a standard 7-segment display generally has eight input connections, one for each LED segment and one that acts as a common terminal or connection for all the internal display segments. Some displays also have a decimal point (DP) option.

Fig. 2.20 shows the pin diagram for IC 74LS47. It comprises of 4 BCD inputs, one pin Lamp Test (LT), one pin Blanking input (RBO), one pin Ripple Blanking Input (RBI), seven outputs (*a – g*), one V_{CC} & one ground.

IC 74LS47 is used to drive common anode 7-segment display and IC 74LS48 for common cathode 7-segment display.

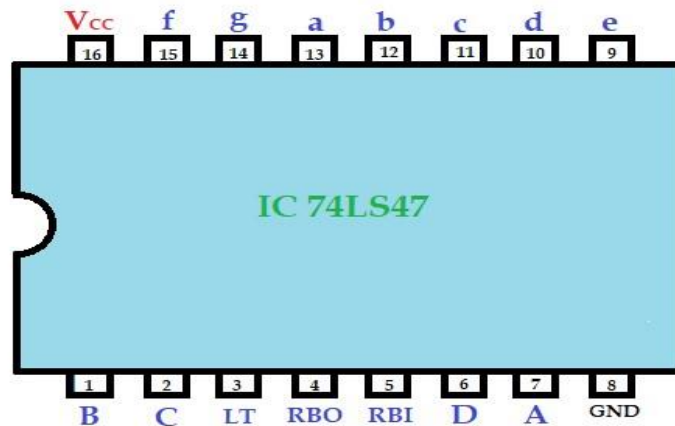


Figure 2.20

Before using the 7-segment display, we need to check it. When Lamp Test LT is LOW and RBO is HIGH, all the segments on the 7-segment display are lit regardless of the inputs D, C, B, and A. If RBO is LOW, all 7 segments are off

so that the display is blank, regardless of any inputs (D, C, B, A). When RBI is low and input DCBA = 0000m the display 0 is blank otherwise the number is displayed. It is used to remove leading zero from a number. This is how to check, if all the LED segments are working.

RBO must be HIGH when output functions 0 to 9 are desired. RBI must be HIGH if blanking of a decimal zero is not desired. The RBO is typically HIGH. If A, B, C, D and RBI are all LOW, and the lamp test LT is HIGH, then all 7 segments are off. In this situation, the RBO goes LOW.

BCD counter follow a sequence of ten states and count using BCD numbers from 0000 to 1001 and then returns to 0000 and repeats. Such a counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits giving a MOD-10 count. To display each number from 0 to 9 requires a decoder circuit, which translates a binary coded number into the appropriate logic levels on each of the display segments.

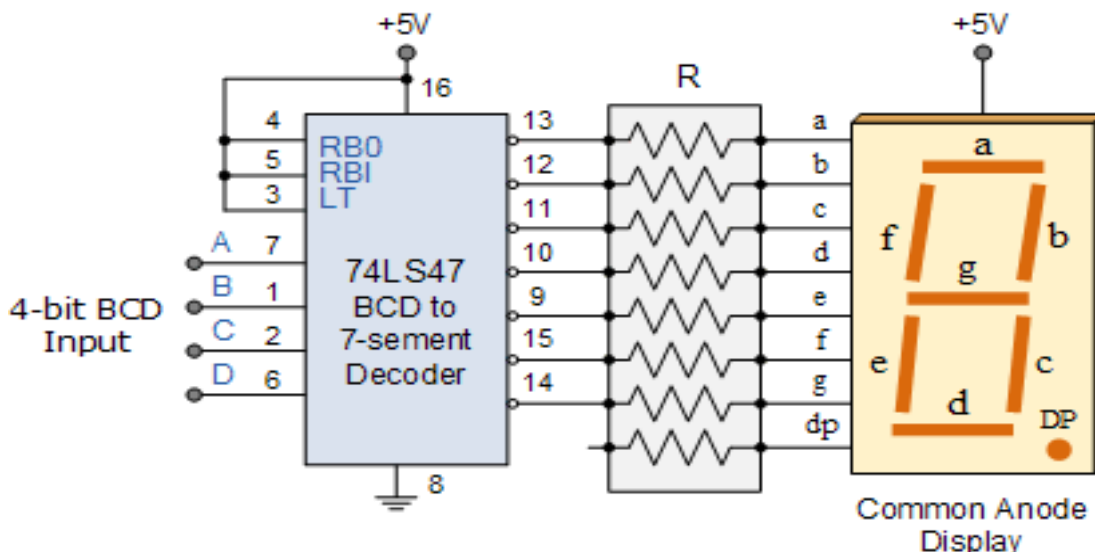


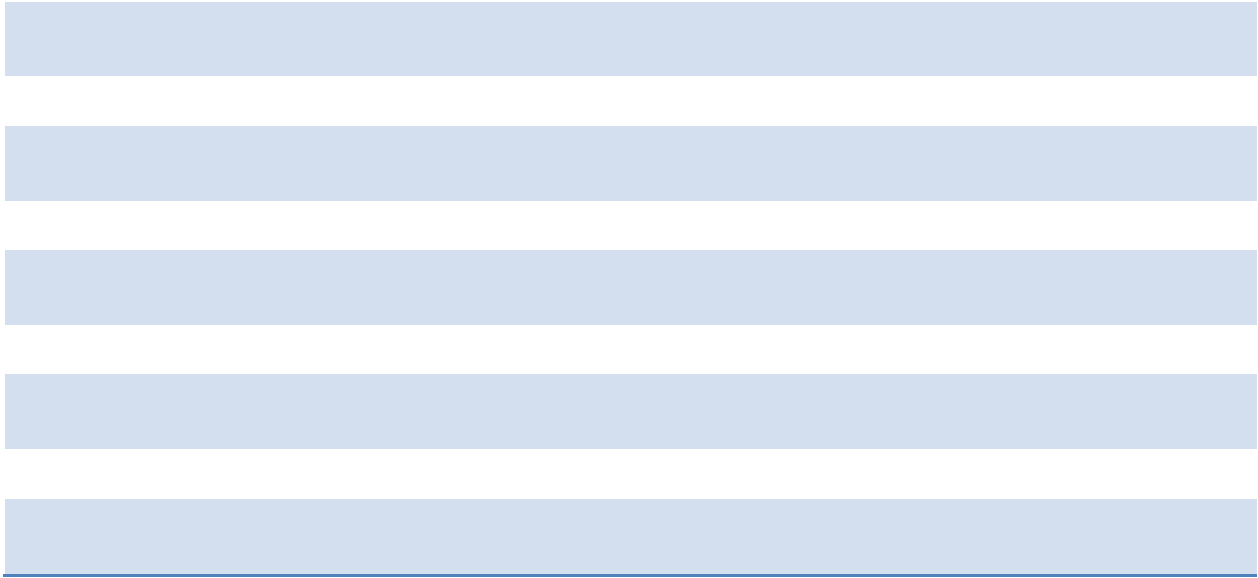
Figure 2.21

Procedure:

- a. Connect LT, RBO, RBI and common pin of seven segment display to 5 V V_{CC} .
- b. Perform Lamp test by keeping LT LOW and observe the segments.
- c. Keeping LT HIGH, give various BCD inputs to D, C, B, A; note the corresponding display. Note the displays for all the possible input combinations.
- d. Keeping inputs DCBA = 0000, note the display. Now make RBI LOW and then note the display.
- e. Repeat step (d) for some other input combination and note the display.

Observations:

INPUTS							DISPLAY
LT	RBO	RBI	D	C	B	A	



Result: The working of seven segment display IC 74LS47/74LS4 has been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} and ground should be applied carefully.

2.6 DIGITAL BUFFER

A digital buffer is a unidirectional device in which the signal passes through them in one direction only from input A to the output at Q as shown in Fig. 2.22. Thus, when input A is at logic 1, output Q is at logic 1, and when input A is at logic 0, output Q is at logic 0.

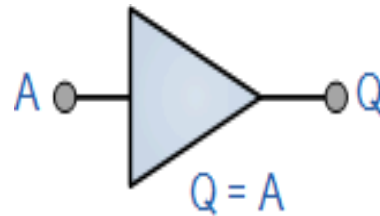


Figure 2.22

Buffers can be used to isolate the interfacing various circuit stages from each other preventing the impedance or operation of one circuit from affecting the impedance or operation of another. Also, on their own, buffers can be used as drivers for high current loads such as transistor switches because their output drive capability (fan out) is generally much higher than their input signal requirements.

The digital non-inverting buffer function can also be made using spare logic AND or logic OR gates or by using pairs of NOT gates as shown in Fig. 2.23.

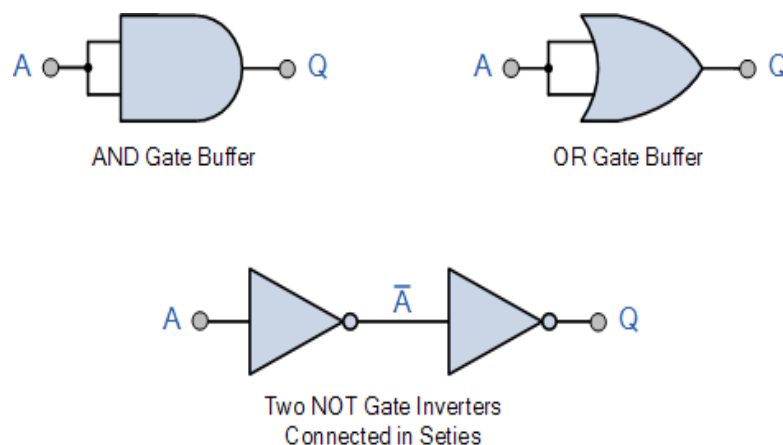


Figure 2.23

One of the disadvantages of a single input digital buffer is that the output at Q will always be at the same logic level as the input, not advisable always. One

way to overcome this is to turn the basic buffer into a 3-state buffer more commonly known as *Tri-State Buffer*.

2.6.1 TRI STATE BUFFER

A Tri-state buffer is another type of buffer circuit which can be used to control the passage of a logic signal from its input to its output. The tri-state buffer is a combination device whose output can be electronically turned ON or OFF by means of an external *Control* or *Enable (EN)* signal allowing them to be used in bus-oriented systems.

As the name implies, the output at Q for a tri-state buffer can take on one of three possible states, logic 0, logic 1 and high - Z (high impedance), that is, an open circuit, rather than the standard 0 or 1 states. The buffers enable can be either a logic 0 or a logic 1. Thus, a tri-state buffer requires two inputs, one being the data input (A) and the other being the control or Enable input (EN) as shown in Fig. 2.24.

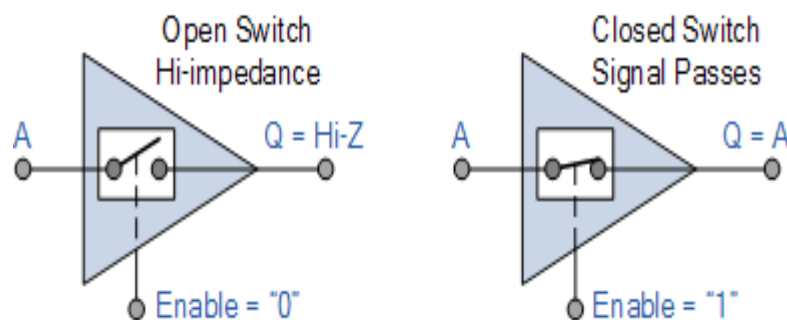


Figure 2.24

The Tri-state buffer's symbol is very similar to the standard buffer symbol shown in Fig. 2.22 but with the addition of a second input representing the enable/disable control function. When the enable (EN) input is at a logic level

1 (for positive logic), it acts as a normal buffer allowing the input signal A to pass directly to the output at Q; whether it is a logic 0 or a logic 1.

When the enable input is at logic 0, the tri state buffer is activated into its third state and disables or turns OFF its output producing an open circuit condition. The third condition is neither at a logic 1 (high) or logic 0 (low), but instead gives an output state that is at a very high impedance, High Z, more commonly written as: Hi-Z.

Thus, a tri state buffer has two logic inputs, 0 or a 1 but can produce three different output states, 0, 1 or Hi - Z which is why it is called a *Tri-state* device. Note that this third state is *not* equal to a logic level 0 or a 1, but is a high impedance state as its output is electrically disconnected.

Then we can correctly state for a positively enabled tri state buffer that:

- If the enable signal is at logic 1, the input signal of the buffer gate is passed directly to its output.
- If the enable signal is at logic 0, the output of the buffer gate acts like an open circuit, that is, high impedance, (Hi - Z).

Experiment 2.6.1: To study IC 74244 as a unidirectional buffer.

Apparatus: IC 74244, 5 V power supply, connecting wires, bread board, DMM.

Theory: IC 74LS244 is a 20 pin Octal Tri-state Buffer as shown in Fig. 2.25.

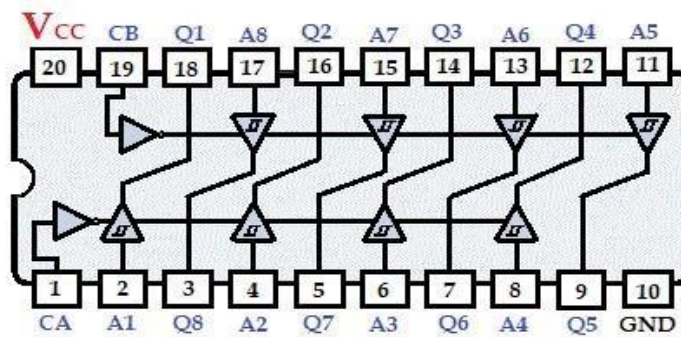


Figure 2.25

The eight buffers are configured into two groups of four with the first group (A1 to A4) being controlled by enable input CA and the second group (A5 to A8) being controlled by the enable input CB. The IC 74LS244 has very high sink and source current capabilities if required to switch to transistor loads.

Procedure:

(A) First Group Buffer (A1 to A4):

- Bread board the circuit as shown in Fig. 2.25 appropriately.
- To enable the first group of buffers, connect CA to ground.
- Apply various four input conditions and note the corresponding outputs.
- Now connect CA to V_{CC} and observe the output.

(B) Second Group Buffer (A5 to A8):

- To enable the first group of buffers, connect CB to ground.
- Apply various four input conditions and note the corresponding outputs.
- Now connect CB to V_{CC} and observe the output.

Observations:

(A) First Group Buffer (A1 to A4):

CE		INPUTS			OUTPUTS			
CA	A4	A3	A2	A1	Q4	Q3	Q2	Q1
L	1	1	0	0				
L	0	0	1	1				
L	0	0	0	0				
L	0	1	0	1				
H	1	0	1	0				

(B) Second Group
Buffer (A5 to A8):

CE		INPUTS			OUTPUTS			
CB	A8	A7	A6	A5	Q8	Q7	Q6	Q5
L	1	1	0	0				
L	0	0	1	1				
L	0	0	0	0				
L	0	1	0	1				
H	1	0	1	0				

Result: The working of IC 74244 as a unidirectional buffer has been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.

2.6.2 MULTIPLE TRI - STATE BUFFERS ON A SINGLE BUS

In Fig. 2.26, the two tri-state buffers are connected in parallel but in reverse, EN acting more like a directional control signal thus allowing data to be read both, *from* and transmitted *to* the same data terminal.

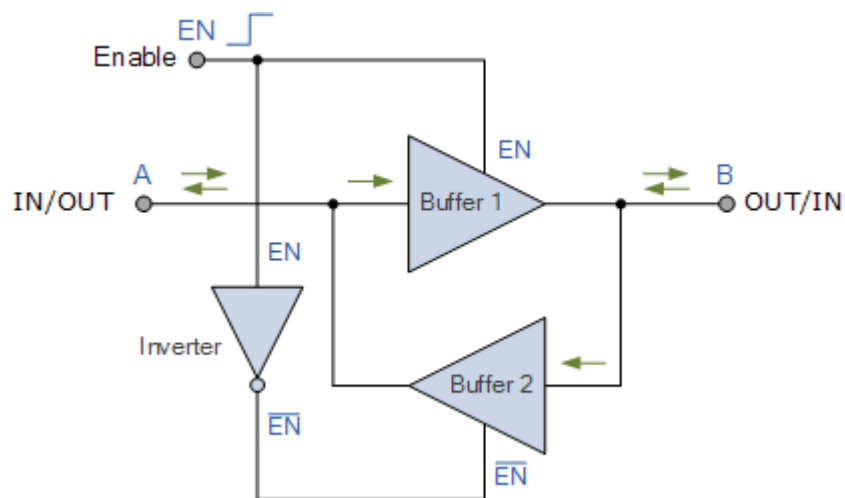


Figure 2.26

When the enable input is HIGH, (EN equals logic 1) data is allowed to pass from A to B via buffer 1, and when the enable input is LOW, (EN equals logic 0) data passes from B to A via buffer 2. Thus, the enable input EN acts as

direction control allowing data to flow in either direction depending upon the logic status of this control input. In this type of application a tri-state buffer with bi-directional switching capability such as the TTL 74LS245 can be used producing what is called a data *Bus Transceiver*.

Experiment 2.6.2: To study IC 74245 as an octal transceiver bidirectional buffer.

Apparatus: IC 74245, 5 V power supply, connecting wires, bread board, DMM.

Theory: IC 74LS245 is a 20 pin Octal Tri - state bidirectional Buffer as shown in Fig. 2.27.

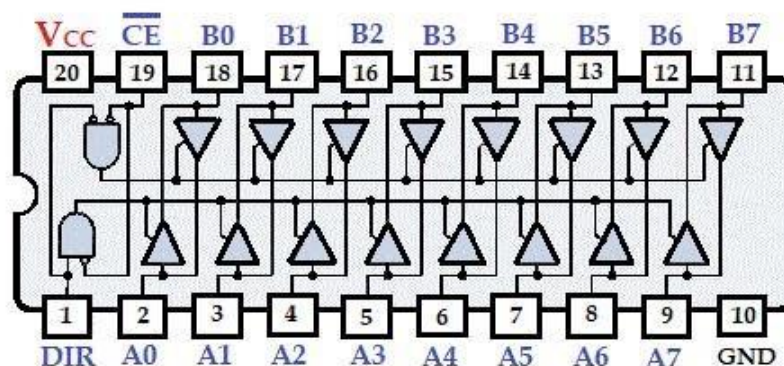


Figure 2.27

The TTL 74LS245 is an octal bus transceiver (Transmitter/Receiver) designed for asynchronous two-way communication between two data buses or input/output device. The transceiver allows for the transmission of data from the terminals A to terminals B or the reverse depending on the logic level at the direction control (DIR) input, (pin 1).

If the direction control input is HIGH at logic level 1, then data will pass from terminal set A to terminal set B. If the direction control input is LOW at logic level 0, then data will pass in the reverse direction from terminal set B to terminal set A.

So when held HIGH at logic level 1, the output chip enable (CE) input, (pin 19) can be used to disable the device so that the terminals, and therefore any connected data buses are effectively isolated from each other in a Hi - Z state.

Procedure:

(A) For inputs A0 to A7:

- a. Bread board the circuit as shown in Fig. 2.27; including V_{CC} and GND.
- b. To enable the IC, connect enable pin to ground.
- c. Keeping DIR at 5 V, A0 to A7 act as inputs and note the corresponding outputs at B0 to B7.
- d. Apply various input conditions and note down the corresponding outputs.

(B) For inputs B0 to B7:

- a. To enable the IC, connect enable pin to ground.
- b. Keeping DIR at 0 V, B0 to B7 act as inputs and note the corresponding outputs at A0 to A7.
- c. Apply various input conditions and note down the corresponding outputs.

Result: The working of IC 74245 as a bidirectional buffer has been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.
3. Same power supply the one used to energize the IC can be used appropriately for inputs & control pins.
4. Based on the IC type sometime enable pin is active low & sometime active high.
5. The O/P can be verified either using LED preceded by a resistor or on DMM.
6. Until the IC is used as an interfacing component in a bigger circuit, the BUFFER action can be verified using few inputs statistically.

2.7 WHAT IS A LATCH?

In digital IC's, the data can be transmitted as well as stored effectively with the help of logic gates. Generally, the latches and flip flops can be used to store one bit of data. These are the building blocks and works like basic elements in computers, electronic systems, etc.

A latch is a device which has two stable states namely high output and low output. This includes a feedback lane; accordingly, data can be stored with the device. A latch is a memory device used to store one bit of data. These are same like flip flops, however, they are not synchronous devices. They do not work on edges of the clock as flip flops do. **Usually LATCH is used when a faster intelligence is interfaced with slower peripheral.**

2.7.1 D - LATCH

The D-latch is used to capture or 'latch' the logic level which is present on the data line when the clock input is high. If the data on the D line changes state while the clock pulse is high, then the output, Q follows the input, D . When the CLK input falls to logic 0, the last state of the D input is trapped and held in the latch.

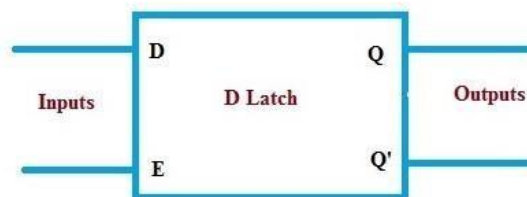


Figure 2.28

Fig. 2.29 shows the timing diagram of a D - latch.

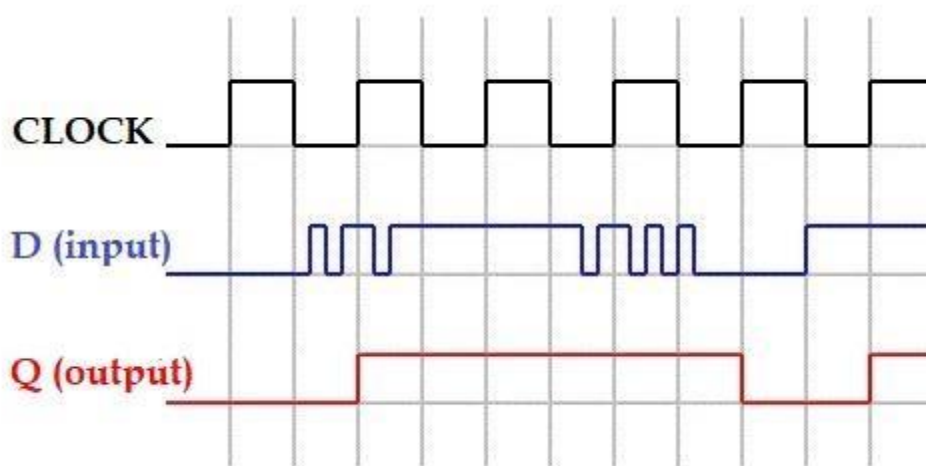


Figure 2.29

From the timing diagram, it is clear that the output Q 's waveform resembles that of input D 's waveform when the clock is high whereas when the clock is low Q retains the previous value of D (the value before clock dropped down to 0).

Experiment 2.7: To study the working of D - Latch IC 74LS373.

Apparatus: IC 74LS373, 5 V power supply, bread board, DMM, connecting wires.

Theory: IC 74LS373 high speed octal D -type latch utilizes advanced silicon gate CMOS technology. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS - TTL loads. Due to the large output drive capability and the 3-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

Features:

- Typical propagation delay: 18 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μA maximum
- Low quiescent current: 80 μA maximum (74 Series)
- Output drive capability: 15 LS - TTL loads

Fig. 2.30 shows the pin diagram of IC 74LS373. It is a 20 pin IC comprising of eight input lines (D_0 - D_7) and eight output lines (O_0 - O_7). The OE pin is meant for output enable and LE pin for latch enable. And the rest two pins V_{CC} for power supply and GND for ground.

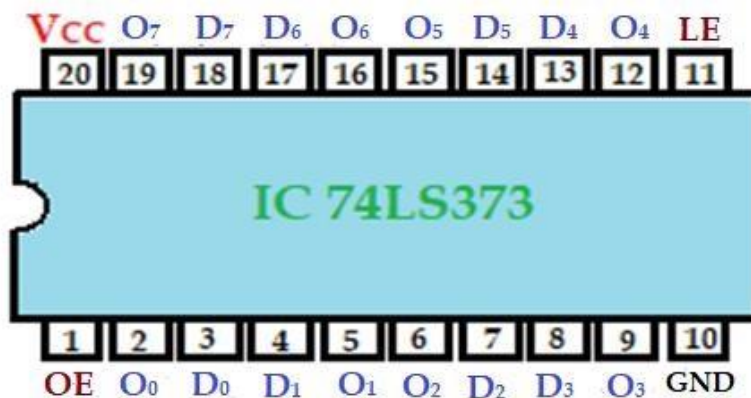


Figure 2.30

When the LATCH ENABLE (LE) input is HIGH, the Q outputs will follow the D inputs. When the LE goes LOW, data at the D inputs will be retained at the outputs until LE returns to HIGH again.

The OUTPUT ENABLE (OE) pin also plays an important role in the working of this IC. When the OE pin is LOW, input data will appear in the output. When a HIGH logic level is applied to the OE input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Procedure:

- a. Bread board the circuit as shown in Fig. 2.30 with making appropriate connections for V_{CC} and GND.
- b. Keeping OE as LOW and LE as HIGH, apply various data inputs varying from $D7$ to $D0$ and note down the outputs using DMM.
- c. Keeping OE as LOW and LE as LOW, apply various data inputs varying from $D7$ to $D0$ and note down the outputs using DMM.
- d. Keeping OE as HIGH and LE as X (either HIGH or LOW), apply various data inputs varying from $D7$ to $D0$ and note down the outputs using DMM.

Observations:

Output Enable (OE)	Latch Enable (LE)	Data Input D7 to D0	Output (V)							
			O7	O6	O5	O4	O3	O2	O1	O0
L	H	11111111								
L	H	00000000								
L	H	01100101								
L	L	00101011								
L	L	11011110								
H X	01101101									

Result: The working of D - Latch IC 74LS373 with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. Same power supply the one used to energize the IC can be used appropriately for inputs & control pins.
3. V_{CC} and ground should be applied carefully at the specified pins only.
4. The O/P can be verified either using LED preceded by a resistor or on DMM.

5. Until the IC is used as an interfacing component in a bigger circuit, the LATCH action can be verified using few inputs statistically.

2.8 SHIFT REGISTER

The Shift register is another type of sequential logic circuit that can be used for the storage or the transfer of binary data. This sequential device loads the data present on its input and then moves or *shifts* it to its output once every clock cycle, hence the name *Shift Register*.

A *shift register* basically consist of several single bit D-type Data Latches, one for each data bit, either a logic 0 or a 1, connected together in a serial type daisy chain arrangement so that the output from one data latch becomes the input of the next latch and so on.

Data bit may be fed in or out of a shift register serially, that is one after the other from either the left or the right direction or all together at the same time in a parallel configuration.

The number of individual data latches required to make up a single *Shift Register* device is usually determined by the number if bits to be stored with the most common being 8-bits (one byte) wide constructed from eight individual data latches.

Shift Registers are used for data storage or for the movement of data and are therefore commonly used inside calculators or computers to store data such as two binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format. The individual data latches

that make up a single shift register are all driven by a common clock (Clk) signal making them synchronous devices.

Shift register IC's are generally provided with a *clear* or *reset* connection so that they can be SET or RESET as required. Generally, shift registers operate in one of four different modes with the basic movement of data through a shift register being:

- **Serial - in to Parallel - out (SIPO):** the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.
- **Serial - in to Serial - out (SISO):** the data is shifted serially IN and OUT of the register, one bit at a time in either a left or right direction under clock control.
- **Parallel - in to Serial - out (PISO):** the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
- **Parallel - in to Parallel - out (PIPO):** the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

The effect of data movement from left to right through a shift register is represented in Fig. 2.31.

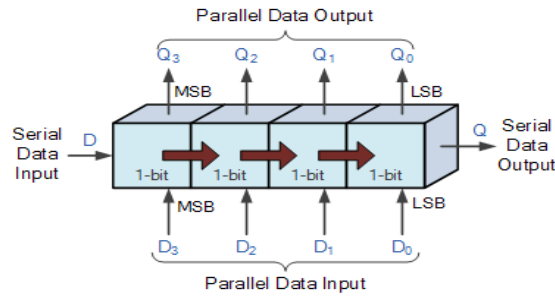


Figure 2.31

Also, the directional movement of the data through a shift register can be either to the left (left shifting) to the right, (right shifting) left in but right out, (rotation) or both left and right shifting within the same register thereby making it bidirectional. Here, it is assumed that all the data shifts to the right, (right shifting).

2.8.1 4 - BIT SERIAL - IN TO PARALLEL - OUT SHIFT REGISTER

Let us assume that all the flip flops (FFA-FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level 0 i.e., no parallel data output. This arrangement is shown in Fig. 2.32.

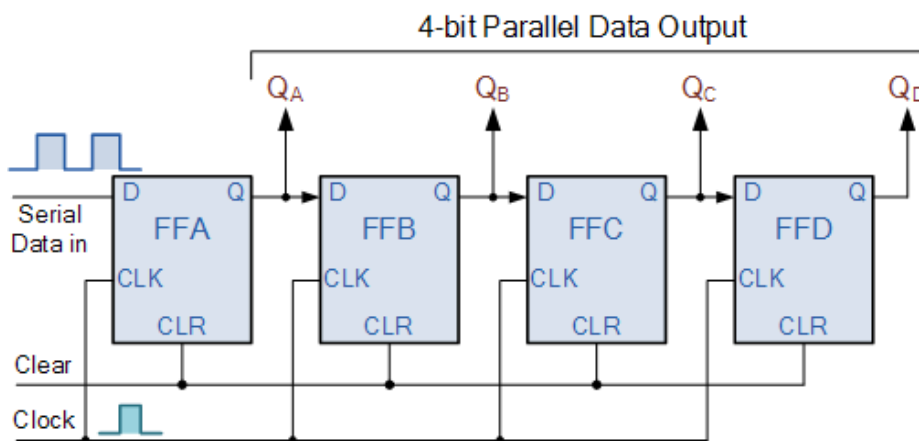


Figure 2.32

If a logic 1 is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting Q_A will be set HIGH to logic 1 with all the other outputs still remaining LOW at logic 0. Assume now that the DATA input pin of FFA has returned LOW again to logic 0 giving us one data pulse or 0 - 1 - 0.

The second clock pulse will change the output of FFA to logic 0 and the output of FFB and Q_B HIGH to logic 1 as its input D has the logic 1 level on it from Q_A . The logic 1 has now moved or been shifted one place along the register to the right as it is now at Q_A .

When the third clock pulse arrives this logic 1 value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level 0 because the input to FFA has remained constant at logic level 0.

The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the Table 2.1 until the complete data value of 0 - 0 - 0 - 1 is stored in the register. This data value can now be read directly from the outputs of Q_A to Q_D .

Table 2.1

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0

3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

Then the data has been converted from a serial data input signal to a parallel data output. Fig. 2.33 shows the waveform propagation of the logic 1 through the register from left to right.

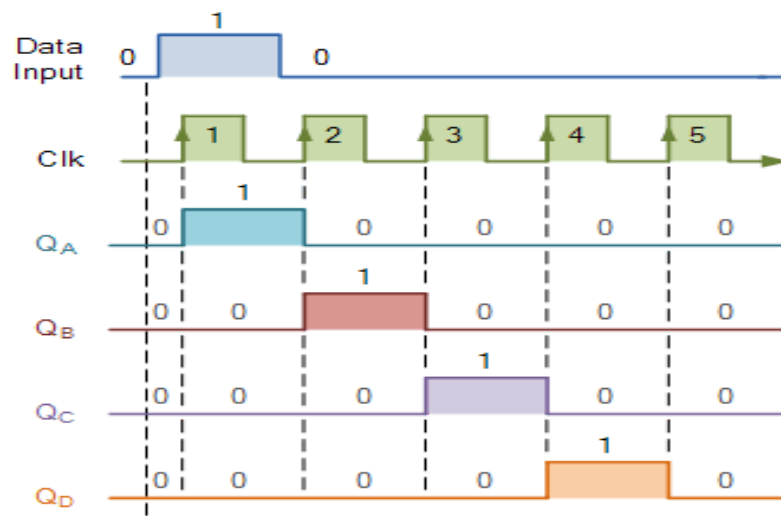


Figure 2.33

After the fourth clock pulse has ended the 4 - bits of data (0 - 0 - 0 - 1) are stored in the register and will remain there provided clocking of the register has stopped. In practice, the input data to the register may consist of various combinations of logic 1 and 0. Commonly available SIPO IC's include the standard 8-bit 74LS164 or the 74LS594.

2.8.2 4 - BIT SERIAL - IN TO SERIAL - OUT SHIFT REGISTER

This shift register is very similar to the SIPO, except where before the data was read directly in a parallel form from the outputs Q_A to Q_D , this time the data is allowed to flow straight through the register and out of the other end. Since there is only one output, the serial DATA leaves the shift register one bit at a time in a serial pattern, hence the name *Serial - in to Serial - out Shift Register* or SISO.

The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip flop, the serial output (SO) which is taken from the output of the right hand flip flop and the sequencing clock signal (CLK). Fig. 2.34 shows a generalized serial - in serial - out shift register.

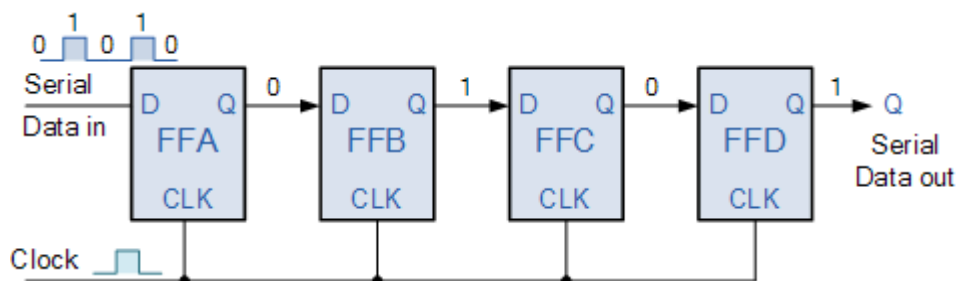


Figure 2.34

This type of shift register also acts as a temporary storage device or it can act as a time delay device for the data, with the amount of time delay being controlled by the number of stages in register, 4, 8, 16, etc. or by varying the application of clock pulses. Commonly available IC's include the 74HC595 8-bit Serial - in to Serial - out Shift register all with 3 - state outputs.

2.8.3 4 - BIT PARALLEL - IN TO SERIAL - OUT SHIFT REGISTER

The Parallel - in to Serial - out shift register acts in the opposite way to the serial - in to parallel out. The data is loaded into the register in a parallel format in which all the data bits enter their inputs simultaneously, to the parallel input pins P_A to P_D of the register. The data is then read out sequentially in the normal shift right mode from the register at Q representing the data present at P_A to P_D as shown in Fig. 2.35.

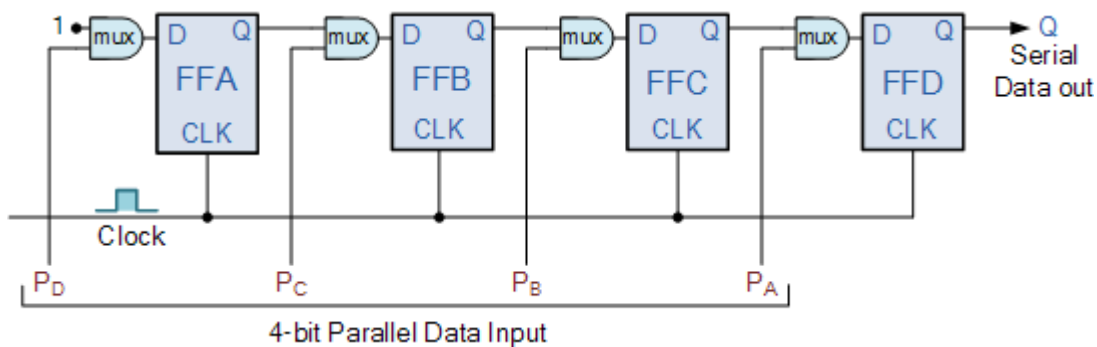


Figure 2.35

This data is outputted one bit a time on each clock cycle in a serial format. It is important to note that with this type of data register a clock pulse is not required to parallel load the register as it is already present, but four clock pulses are required to unload the data.

As this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line. Commonly available IC's include the 74HC166 8 - bit Parallel - in Serial Out Shift Registers.

2.8.4 4 - BIT PARALLEL - IN TO PARALLEL - OUT SHIFT REGISTER

The final mode of operation is the Parallel - in to Parallel - out Shift Register. This type of shift register also acts as a temporary storage device or as a time delay device similar to the SISO configuration. The data is presented in a parallel format to the parallel input pins P_A to P_D and then transferred together directly to their respective output pins Q_A to Q_D by the same clock pulse. Then one clock pulse loads and unloads the register as shown in Fig. 2.36.

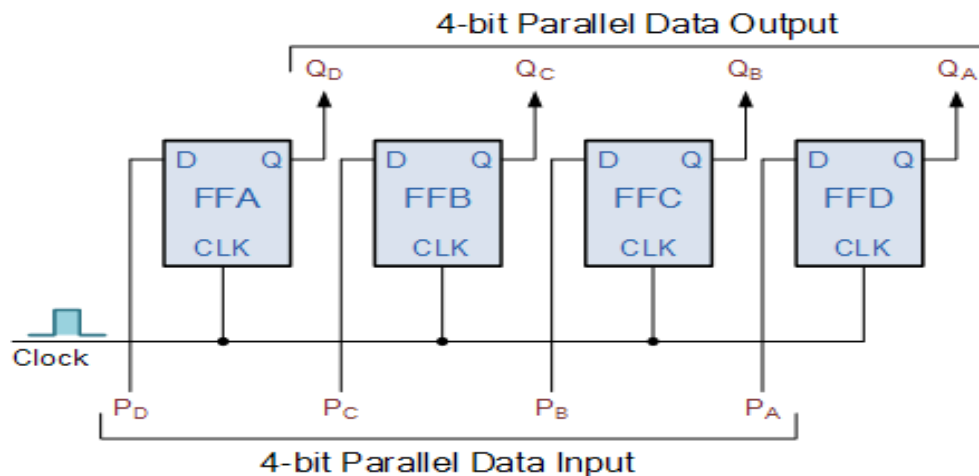


Figure 2.36

The PIPO shift register is the simplest of the four configurations as it has only three connections, the parallel input (PI) which determines what enter the flip flop, the parallel output (PO) and the sequencing clock signal (CLK).

Similar to SISO shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the frequency of the clock pulses. Also, in this type of register there are no inter connections between the individual flip flops since no serial shifting of the data is required.

2.8.5 UNIVERSAL SHIFT REGISTER

Today, there are many high speed bi-directional *universal* type Shift Registers available such as the TTL 74LS194, 74LS195 or the CMOS 4035 which are available as 4 - bit multi - function devices that can be used in either serial - to - serial, left shifting, right shifting, serial - to - parallel, parallel - to - serial, or as a parallel - to - parallel multifunction data register, hence the name *Universal*.

These universal shift registers can perform any combination of parallel and serial input to output operations but require additional inputs to specify desired function and to preload and reset the device. Fig. 2.37 shows a commonly used universal shift register TTL 74LS194.

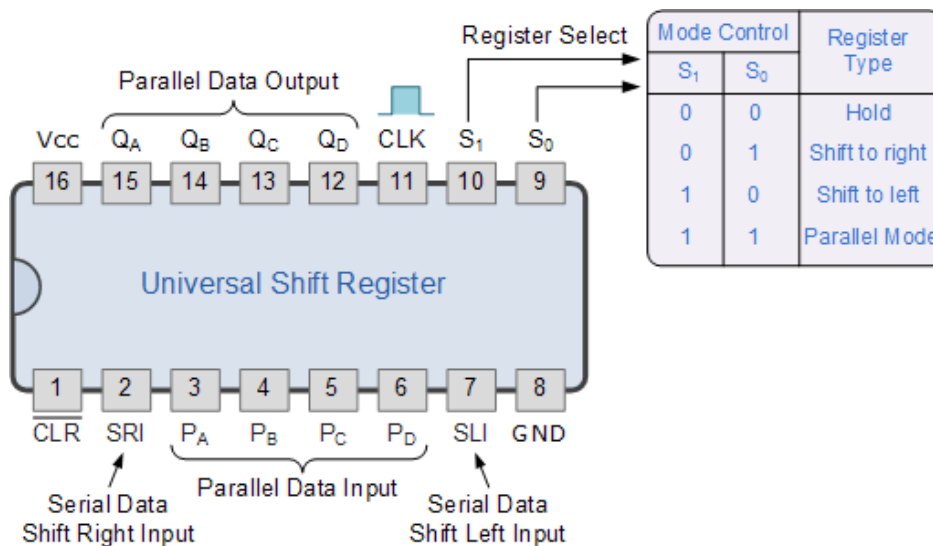


Figure 2.37

Universal shift registers are very useful digital devices. They can be configured to respond to operations that require some form of temporary memory storage

or for the delay of information such as the SISO or PIPO configuration modes or transfer data from one point to another in either a serial or parallel format. Universal shift registers are frequently used in arithmetic operations to shift data to the left or right for multiplication or division.

Experiment 2.8: To study the working of Shift Register IC 74LS95 as a serial Shift and parallel - in parallel - out register.

Apparatus: IC 74LS95, IC 7400 (NAND gate), 1 K Ω resistors, 5 V power supply, bread board, DMM, connecting wires, LEDs.

Theory: IC 74LS95 is a 4 – bit Shift Register with serial and parallel synchronous operating modes. The serial shift right and parallel load are activated by separate clock inputs which are selected by a mode control input. The data is transferred from the serial or parallel inputs to the Q outputs synchronous with the HIGH to LOW transition of the appropriate clock input.

Features:

- Synchronous, expandable Shift Right
- Synchronous Shift Left Capability
- Synchronous Parallel Load
- Separate Shift and Load Clock Inputs
- Supply Voltage: 4.5 – 5.5 V
- Operating Ambient Temperature Range: - 55 °C to 125 °C

Fig. 2.38 shows the pin diagram of IC 7495. It is a 14 - pin IC comprising of one serial input, 4 inputs (A, B, C, D), one mode control pin, two clock pins (Clock - 1 R - shift and Clock - 2 L - shift), 4 outputs (Q_A , Q_B, Q_C, Q_D), one V_{CC} and one ground.

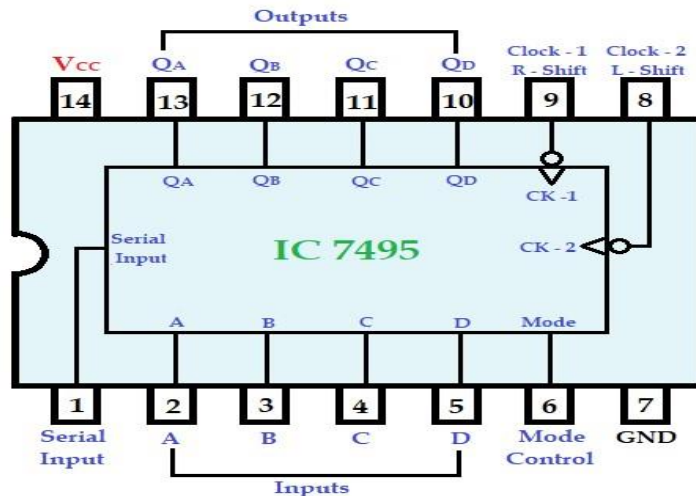


Figure 2.38

The left and right shifts are operated by separate clocks which are selected by a mode control pin. The R - shift is operated by clock 1 and left shift is operated by clock 2.

De-bounce Circuit:

Fig. 2.39 shows a debounce circuit with NAND gates. The two digital logic gates are connected as a pair of cross coupled NAND gates with active LOW inputs forming a SR latch circuit as two of the NAND gate inputs are held HIGH by the two 1 KΩ pull up resistors.

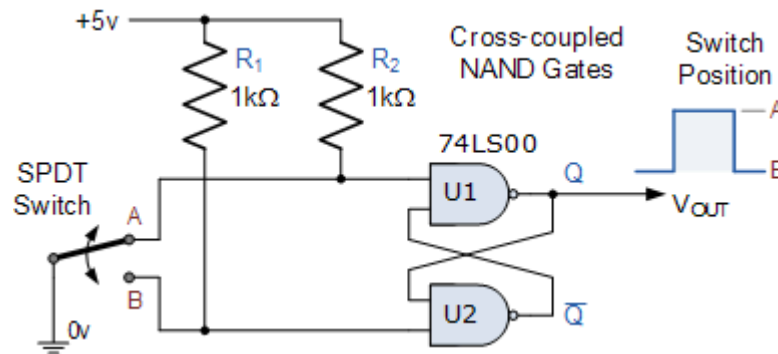


Figure 2.39

When the switch of the cross coupled NAND de-bounce circuit is in position A, NAND gate U1 is *set* and the output at Q is HIGH at logic 1. When the switch is moved to position B, U2 becomes *set* which resets U1. The output at Q is now LOW at logic 0.

The operating switch between positions A and B toggles or switches the output at Q from HIGH to LOW or from LOW to HIGH. As the latch requires two switching actions to set and reset it, any bouncing of the switch contacts in either direction for both opening and closing are not seen at the output Q. Also, the advantage of this SR latch de-bounce circuit is that it can provide complementary outputs at Q and \bar{Q} .

Procedure:

1. Construct the debounce circuit as shown in Fig. 2.39 using IC 7400. This circuit will be used as a clock generator (manual clock) i.e., the output of this circuit will be given as clock pulses to the clock input of the shift register.

2. When the switch (GND, 0 V) is connected to A, positive half cycle of the square wave is formed and when switch (GND, 0 V) is connected to B, negative half cycle of the square wave is formed.

(A) For Parallel – In Parallel Out:

1. Bread board the circuit as shown in Fig. 2.38 with making appropriate connections for V_{CC} and GND.
2. Apply the four bit data at A, B, C, D.
3. Keeping Mode control at HIGH, apply one clock pulse to CLK – 2.
4. Observe the corresponding outputs simultaneously at Q_A , Q_B , Q_C , Q_D during HIGH to LOW transition of the clock.
5. The Serial input and CLK – 1 can be either kept open or connected to ground since they are in don't care mode.

(B - 1) For Serial-in Serial-out (Shift Right Operation):

1. For serial shift right operation, keep the Mode control pin at LOW.
2. Load the shift register with 4 bits of data one by one serially. The input data (D_S) is given to the Serial Input to Q_A .
3. At the end of the 4th clock pulse first data D_0 appears at Q_D . With another clock pulse, next data D_1 will appear at Q_D . After the application of another clock pulse, data D_2 will appear at Q_D . In the next clock pulse, data D_3 appears at Q_D .
4. The CLK – 2 pin is kept in don't care mode by making it grounded or left open.

(B - 2) For Serial – In Serial Out (Shift Left Operation):

1. For serial shift right operation, keep the Mode control pin at HIGH, the Serial input pin and CLK – 1 should be kept in don't care mode.

2. Keeping pins A and Q_A , B and Q_B , C and Q_C shorted; apply the serial input to D.
3. At the end of the 4th clock pulse first data D_0 appears at Q_A . With another clock pulse, next data D_1 will appear at Q_A . After the application of another clock pulse, data D_2 will appear at Q_A . In the next clock pulse, data D_3 appears at Q_A .

Observations:

(A) For Parallel - In Parallel Out:

Clock Pulse	INPUTS				OUTPUTS			
	A	B	C	D	Q_A	Q_B	Q_C	Q_D
1								
2	D =							
3	B =							
4	A =							
5	C =							

(B - 1) For Serial -in Serial-out (Shift Right Operation):

Clock Pulse	Serial Input D_s	OUTPUTS			
		Q_A	Q_B	Q_C	Q_D
0	X				
1	a =				
2	b =				

3	c =				
4	d =				
5	e =				

* Here a, b, c, d, e, are bits having value either 0 or 1.

(B - 2) For Serial - In Serial Out (Shift Left Operation):

Clock Pulse	Serial Input D	OUTPUTS			
		Q _A	Q _B	Q _C	Q _D
0	X				
1	a =				
2	b =				
3	c =				
4	d =				
5	e =				
6	f =				

* Here a, b, c, d, e, are bits having value either 0 or 1.

Result: The working of Shift Register IC 74LS95 as a serial shift and parallel - in parallel - out register with various inputs have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & the V_{CC} should be applied carefully.
3. **The actual use of shift register is when 8-Bit parallel data is to be sent from one machine to another machine through single cable (parallel to serial) and then to 8-Bit parallel data (serial to parallel) for the machine at the other end.**

2.9 COUNTERS

A counter is probably one of the most useful and versatile subsystems in a digital system. A counter driven by a clock can be used to count the number of clock cycles.

Counters usually counts in binary, can be converted into decimal using BCD decoder and can be displayed on a seven segment display.

Since the clock pulses occur at known intervals, the counter can be used as an instrument for measuring time period or frequency. There are basically two different types of counters; synchronous & asynchronous.

The ripple counter is simple and straight forward in operation and construction and usually requires a minimum of hardware. However, have a speed limitation. Each flip flop is triggered by the previous flip flop and thus the counter has a cumulative settling time. Counters such as these are called *serial* or *asynchronous*.

An increase in speed of operation can be achieved by use of a *parallel* or *synchronous* counter. Here, every flip flop is triggered by the clock (in synchronism) and thus settling time is simply equal to the delay time of a single flip flop. The increase in speed is usually obtained at the price of increased hardware.

2.9.1 RIPPLE COUNTER

A binary ripple counter can be constructed by use of clocked JK flip flop. Fig. 2.40 shows four master slave, JK flip flops connected in a cascade. The system clock, a square wave, drives flip flop A. The output of A drives B, the output of B drives C and the output of C drives flip flop D. All the J and K inputs are tied to V_{CC}. This means that each flip flop will change state (toggle) with a negative transition at its clock pulse.

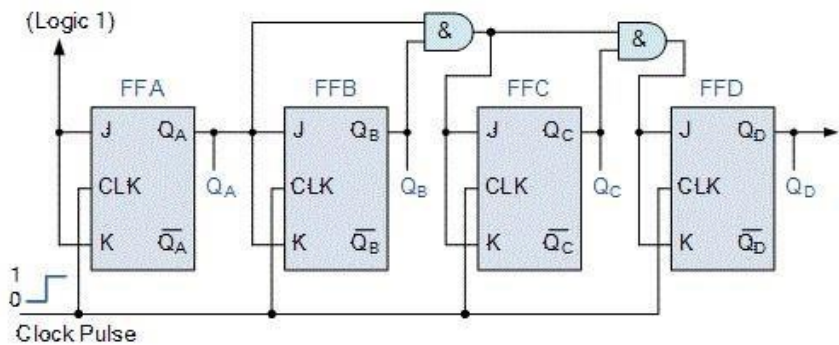


Figure 2.40

Fig. 2.41 shows the timing diagram of a 4 bit ripple counter.

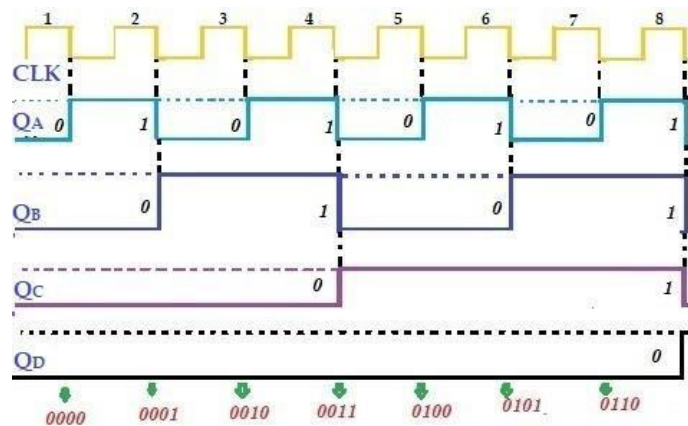


Figure 2.41

The total number of counts that a counter can count is called its *Modulus*. A counter that returns to zero after n counts is called a *modulo- n counter*, for example, a modulo-8 (MOD-8) or modulo-16 (MOD-16) counter, etc. And for an n -bit counter, the full range of the count is from 0 to $2^n - 1$.

Experiment 2.9.1: To study the working of 4-bit Ripple Counter using IC 7493.

Apparatus: IC 74LS93, IC 7400 (NAND gate), 1 K Ω resistors, 5 V power supply, bread board, DMM, connecting wires, LEDs.

Theory: IC 74LS93 is a 4 - bit binary counter composed of 4 JK Flip Flop. Out of these 4 flip flops, 3 are connected together and one is alone. It consists of a MOD-8 counter and a MOD-2 counter, which can be combined together to make a MOD - 16 counter.

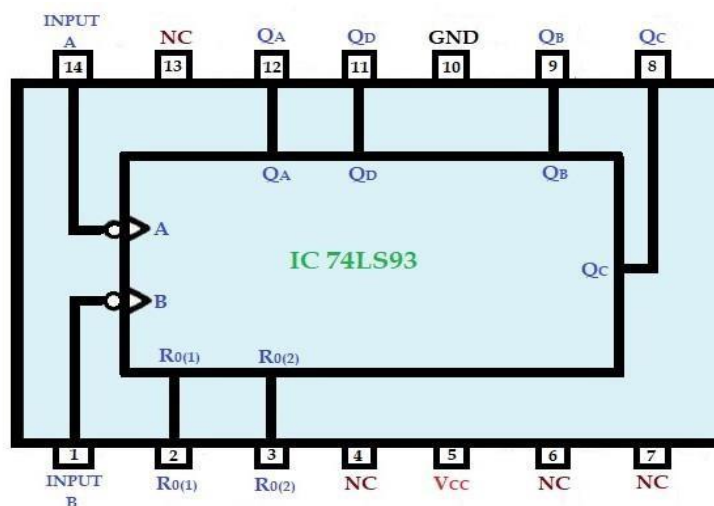


Figure 2.42

Fig. 2.42 shows the pin diagram of IC 74LS93. It consists of two clock inputs (CLK A and CLK B), 4 output pins (Q_A , Q_B , Q_C , Q_D), one Reset-Clear input R, one V_{CC} and one ground. CLK A is the clock input for the first counter and CLK B is the clock input for the second counter. R0 (1) and R0 (2) are the two reset pins of the counter. They are connected to the ground to enable the counter IC.

Features:

- The typical operating voltage is 5 V.
- The input clock frequencies can be up to 32 MHz.
- The output High and Low voltages are 3.5 V and 0.25 V.
- This a 14 pin DIP (Dual in - Line) package IC.

Procedure:

1. Construct the debounce circuit as shown in Fig. 2.39 using IC 7400. This circuit will be used as a clock generator (manual clock) i.e., the output of this circuit will be given as clock pulses to the clock input of the counter.
2. When the switch (GND, 0 V) is connected to A, positive half cycle of the square wave is formed and when switch (GND, 0 V) is connected to B, negative half cycle of the square wave is formed.

(A) Counter with different Modulus:

1. Bread board the circuit as shown in Fig. 2.42 with making appropriate connections for V_{CC} and GND.
2. Construct the counter for different modulus as mentioned below:

(A - 1) MOD - 2:

1. Connect R0 (1) & R0 (2) to ground.
2. Apply clock pulses to the i/p A & note the corresponding output at Q_A .

(A - 2) MOD - 8:

1. Connect R0 (1) and R0 (2) to ground.
2. Apply the clock pulses to the input B and note the corresponding output at Q_B , Q_C , and Q_D .

(A - 3) MOD - 5:

1. Apply the clock pulse to input B.
2. Connect output Q_B and Q_D to R0 (1) and R0 (2) respectively and note the corresponding output at Q_C , Q_B and Q_A .

(A - 4) MOD - 16:

1. Connect output Q_A to input B.
2. Apply the clock pulses to the input A and note the corresponding output at Q_A , Q_B , Q_C and Q_D .

Observations:

(A - 1) MOD - 2:

Clock Pulse	Q_A
1	
2	

(A - 2) MOD - 8:

Clock Pulse	Q _D	Q _C	Q _B	Q _A
1				
2				
3				
4				
5				
6				
7				
8				

A- 3) MOD - 5:

Clock Pulse	Q _C	Q _B	Q _A
1			
2			
3			
4			
5			

(A - 4) MOD - 16:

Clock Pulse	Q _D	Q _C	Q _B	Q _A
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Result: The working of 4-bit Ripple Counter using IC 7493 for various modulus have been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.
3. **Sometimes if the sequence of A, B, C, D does not work, apply first clock to input of F-F B and output of the F-F D as input of A. Now the sequence will be Q_B, Q_C, Q_D & Q_A respectively.**

2.9.2 BCD COUNTER

A BCD counter is a special type of a digital counter which can count up to ten on the application of a clock signal. A toggle T-type flip flop can be used as individual divide by two counters. If we connect several toggle flip flops together in a series chain we can produce a digital counter which stores or display the number of times a particular count sequence has occurred.

Clocked T-type flip flops acts as a binary divide by two counter and in asynchronous counters, the output of one counting stage provides the clock pulse for the next stage. Then a flip - flop counter has two possible output states and by adding more flip flop stages, we can make a divide by 2^N counter. A 4-bit binary counter counts from 0000 to 1111 i.e., from 0 to 15 in decimal.

To make a digital counter which counts from 1 to 10, we need to have the counter count only the binary numbers 0000 to 1001 i.e., from 0 to 9 in decimal and fortunately for us, counting circuits are readily available as integrated circuit with one such circuit being the *Asynchronous 74LS90 Decade Counter*.

Digital counters count upwards from zero to some predetermined count value on the application of a clock signal. Once the count value is reached, resetting them returns the counter back to zero again.

A decade counter counts in a sequence of ten and then returns back to zero after the count of nine. Obviously to count up to a binary value of nine, the counter must have at least four flip flops within its chain to represent each decimal digit.

Fig. 2.43 shows a BCD counter state diagram. A decade counter has four flip flops and 16 potential states of which only 10 are used.

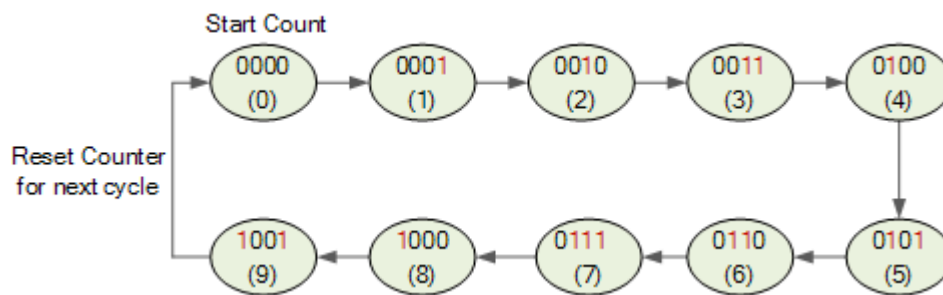


Figure 2.43

It is called a BCD counter because its ten state sequence is that of a BCD code and does not have a regular pattern, unlike a straight binary counter. Hence, a single stage BCD counter such as the 74LS90 counts from decimal 0 to decimal 9 and is therefore capable of counting up to a maximum of nine pulses. Also a digital counter may count up or count down or count up and down (bidirectional) depending on an input control signal.

Experiment 2.9.2: To study the working of BCD Decade Counter using IC 7490.

Apparatus: IC 74LS90, IC 7400 (NAND gate), 1 KΩ resistors, 5 V power supply, bread board, DMM, connecting wires, LEDs.

Theory: The 74LS90 integrated circuit is basically a MOD-10 decade counter that produces a BCD output code. The 74LS90 consists of four master – slave JK flip flops internally connected to provide a MOD-2 (count to 2) counter and a MOD-5 (count to 5) counter. The 74LS90 has one independent toggle JK flip flop driven by the CLK A input and three toggle JK flip flops that form an asynchronous counter driven by the CLK B input as shown in Fig. 2.44.

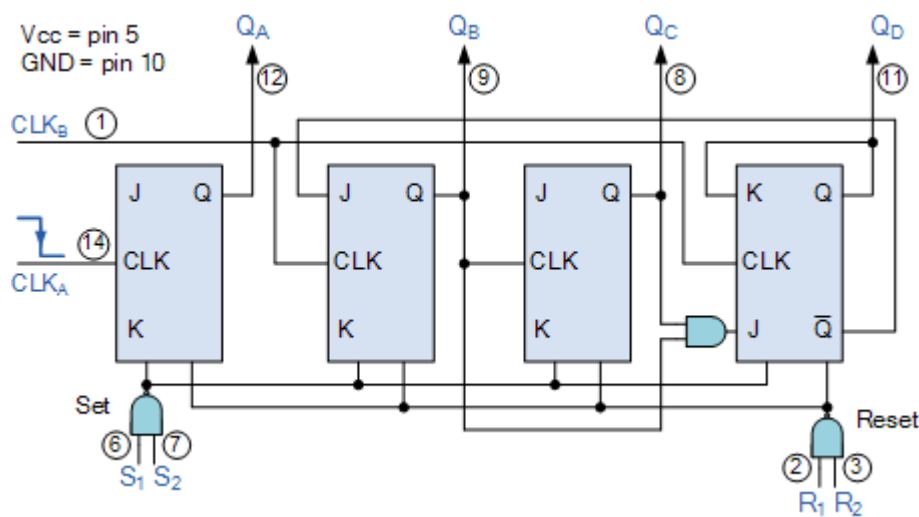


Figure 2.44

The counter's four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD counter circuit code. The 74LS90 counting sequence is triggered on the negative going edge

of the clock signal, that is when the clock signal CLK goes from a logic 1 (HIGH) to logic 0 (LOW).

The additional input pins R_1 and R_2 are counter *reset* pins while inputs S_1 and S_2 are *set* pins. When connected to logic 1, the Reset inputs R_1 and R_2 reset the counter back to zero, 0 (0000) and when the Set inputs S_1 and S_2 are connected to logic 1, they Set the counter to maximum or 9 (1001) regardless of the actual count number or position.

The 74LS90 counter consists of a divide by 2 counter and a divide by five counter within the same package. Then we can use either counter to produce a divide by 2 frequency counter only, a divide by 5 frequency counter only or the two together to produce our desired divide by 10 BCD counter.

With the four flip flops making up the divide by 5 counter section disabled, if a clock signal is applied to CLK A and the output take from Q_A , a standard divide by 2 binary counter can be produced for use in frequency dividing circuits as shown in Fig. 2.45.

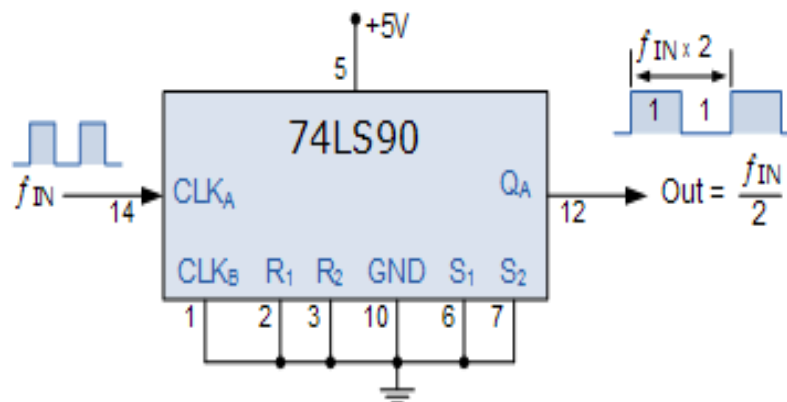


Figure 2.45

To produce a standard divide by 5 counter, disable the first flip flop and apply the clock input signal directly to CLK B with the output signal being taken from Q_D as shown in Fig. 2.46.

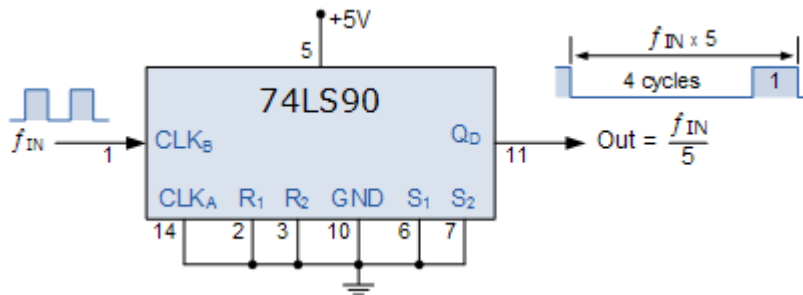


Figure 2.46

In the divide by 5 configuration, the output waveform is not symmetrical but has a 4: 1 mark space ratio. That is four input clock signals creates a LOW or logic 0 output and the fifth input clock signal produces a HIGH or logic 1 output.

To produce a divide by 10 BCD decade counter, both internal counter circuits are used giving a 2 times 5 divide by value. Since the first output Q_A from flip flop A is not internally connected to the succeeding stages, the counter can be extended to form a 4 bit BCD counter by connecting this Q_A output to the CLK B as shown in Fig. 2.47.

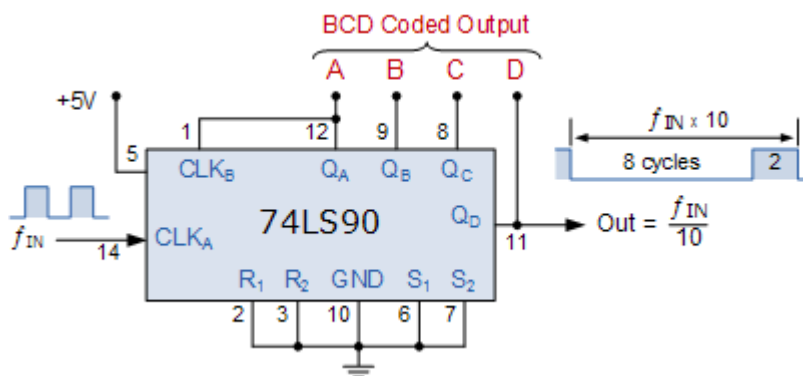


Figure 2.47

Hence BCD counters are binary counters that count from 0000 to 1001 and then resets as it has the ability to clear all of its flip flops after the ninth count. If we connect a push button switch (SW_1) to clock input CLK A, each time the push buttons switch is released the counter will count by one. If we connect LED's to the output terminals Q_A , Q_B , Q_C and Q_D , the binary coded decimal count can be viewed as it takes place.

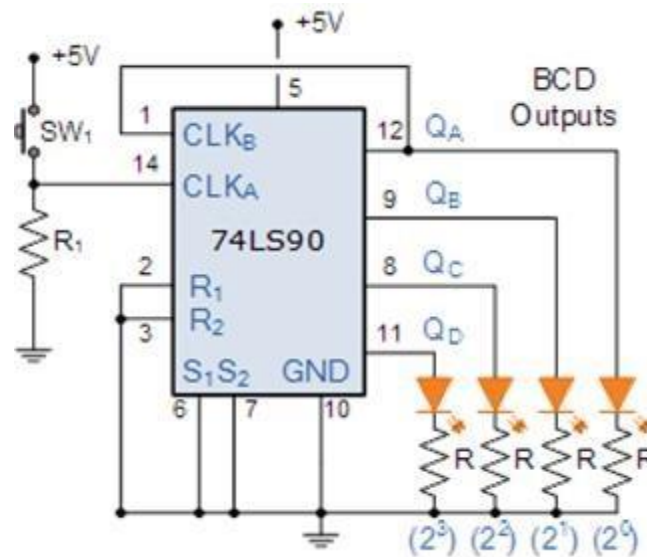


Figure 2.48

Successive applications of the push button switch, SW_1 will increase the count up to nine, 1001. At the tenth application the outputs ABCD will reset back to zero to start a new count sequence. With such a MOD - 10 round number of pulses we can use the decade counter to drive a digital display.

Procedure:

1. Construct the debounce circuit as shown in Fig. 2.39 using IC 7400. This circuit will be used as a clock generator (manual clock) i.e., the output of this circuit will be given as clock pulses to the clock input of the counter.

2. When the switch (GND, 0 V) is connected to A, positive half cycle of the square wave is formed and when switch (GND, 0 V) is connected to B, negative half cycle of the square wave is formed.

(A) Decade Counter with different Modulus:

1. Bread board the circuit with making appropriate connections for V_{CC} and GND.
2. Construct the counter for different modulus as mentioned below:

(A - 1) Divide - by - 2:

1. Make the appropriate connections as shown in Fig. 2.45.
2. Apply the clock pulses to the CLK A and note the corresponding output at Q_A .

(A - 2) Divide by 5:

1. Make the appropriate connections as shown in Fig. 2.46.
2. Apply the clock pulses to the CLK B and note the corresponding output at

Q_D .

(A - 3) Divide by 10:

1. Make the appropriate connections as shown in Fig. 2.47.
2. Apply the clock pulses to the CLK A, also connect the CLK B to the output Q_A and note the corresponding output at Q_D , Q_C , Q_B and Q_A .

Observations:

(A - 1) Divide - by 2:

Clock Pulse	Q _A
1	
2	

(A - 2) Divide - by 5:

Clock Pulse	Q _D	Q _C	Q _B
1			
2			
3			
4			
5			

(A - 3) Divide - by 10:

Clock Pulse	Q _D	Q _C	Q _B	Q _A
0				
1				
2				
3				
4				
5				
6				

7

8

9

10

Result: The working of BCD Decade Counter using IC 7490 has been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. The connections should be tight & V_{CC} should be applied carefully.
3. The count can be displayed using LED as binary counter or using by appropriately connecting 7- segment display as decimal counter (refer appendix A)

2.9.3 PRE - SETTABLE COUNTER

The pre-settable counter is the basic building block that can be used to implement a counter that has any modulus. Nearly all the pre -settable counters available as TTL MSI are constructed by using four flip flops (usually master-slave JK types), and they are generally referred to as 4-bit counters. They may be either synchronous or asynchronous. IC 74193 is a synchronous binary counter which operates in either a count down or count up mode.

Experiment 2.9.3: To study the working of Pre-Settable Counter using IC 74193.

Apparatus: IC 74LS193, IC 7400 (NAND gate), 1 K Ω resistors, 5 V power supply, bread board, DMM, connecting wires, LEDs.

Theory: IC 74193 is a 4-bit pre-settable synchronous MOD-16 UP /DOWN counter IC. It has two separate clock input pins to count up and count down, the output is synchronous with the clock inputs. The separate Terminal count UP & Terminal count DOWN are provided which is useful for designing higher counter or cascading the IC 74193. The Master reset pin is provided to reset the whole IC and also an active low parallel load input pin available to start counting any number.

Fig. 2.49 shows the pin diagram of IC 74193. It comprises of two clock pulse input pins (CPU and CPD), one asynchronous Master Reset input pin (MR), one asynchronous Parallel Load active low input pin (PL), two terminal count output pins (TCU and TCD), four Preset Parallel Data input pins (P_A , P_B , P_C , P_D), four flip flop outputs (Q_A , Q_B , Q_C , Q_D), one V_{CC} and one ground.

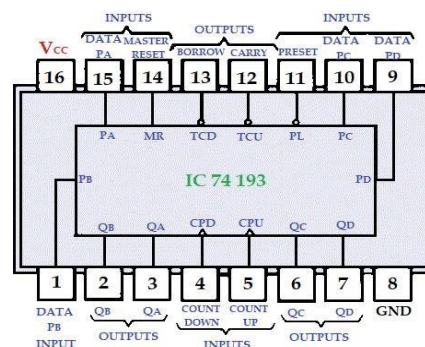


Figure 2.49

In IC 74193, a predefined number/count can be loaded by using its preset pins and supplying a LOW logic to the Parallel Load (PL) pin, hence this IC is known as Pre-settable IC.

For Up counter, the CPU (count up) is connected with clock signals and CPD is connected to logic HIGH. Similarly, for the down counter, the CPD is connected to clock signals and the CPU is connected to logic HIGH.

The Master Reset (MR) terminal is used to reset the IC or to clear the IC output by applying a high signal. A HIGH signal on the Master Reset input will disable the preset gates, override both clock inputs and latch each Q output in the LOW state. If one of the clock inputs is LOW during and after a reset or load operation, the next LOW - to - HIGH transition of that clock will be interpreted as a legitimate signal and will be counted.

The Parallel Load (PL) is an active LOW input pin which is used to load data into the IC. Each circuit has an asynchronous parallel load capability permitting the counter to be preset. When the Parallel Load (PL) and the Master Reset (MR) inputs are LOW, information present on the Parallel Data inputs (P_A, P_B, P_C, P_D) is loaded into the counter and appears on the outputs regardless of the conditions of the clock inputs.

Terminal Count Up (TCU) and Terminal Count Down (TCD) are also active LOW output pins, the output is always HIGH, it goes LOW once IC reaches its maximum and minimum count. When a circuit has reached the maximum count state (15 for the LS193), the next HIGH - to - LOW transition of the Count Up clock will cause

TCU to go LOW. TCU will stay LOW until CPU goes HIGH again, thus effectively repeating the Count Up clock, but delayed by two gate delays. Similarly, the TCD output will go LOW when the circuit is in the zero states and the Count Down clock goes LOW. Since the TC outputs repeat the clock waveforms, they can be used as the clock input signals to the next higher – order circuit in a multistage counter.

Procedure:

1. Construct the de-bounce circuit as shown in Fig. 2.39 using IC 7400. This circuit will be used as a clock generator (manual clock) i.e., the output of this circuit will be given as clock pulses to the clock input of the counter.
2. When the switch (GND, 0 V) is connected to A, positive half cycle of the square wave is formed and when switch (GND, 0 V) is connected to B, negative half cycle of the square wave is formed.

(A) UP Counter:

1. Bread board the circuit as shown in Fig. 2.49 with making appropriate connections for V_{CC} and GND.
2. Ground the parallel data input pins (P_A, P_B, P_C, P_D) and apply the clock pulse to CPU.
3. Apply various inputs combinations at PL and MR, and note the corresponding outputs at Q_A, Q_B, Q_C and Q_D and TCU.

(B) DOWN Counter:

1. Ground the parallel data input pins (P_A, P_B, P_C, P_D) and apply the clock pulse to CPD.
2. Apply various inputs combinations at PL and MR, and note the corresponding outputs at Q_A, Q_B, Q_C and Q_D and TCD.

(C - 1) Pre - Settable UP Counter:

1. Apply various inputs at P_A, P_B, P_C, P_D by making PL pin LOW.
2. Now apply the clock pulse to CPU by making PL pin HIGH
3. Note the corresponding outputs at Q_A, Q_B, Q_C and Q_D .

(C - 2) Pre - Settable DOWN Counter:

1. Apply various inputs at P_A, P_B, P_C, P_D by making PL pin LOW.
2. Now apply the clock pulse to CPD by making PL pin HIGH
3. Note the corresponding outputs at Q_A, Q_B, Q_C and Q_D .

Observations:

(A) UP Counter:

MR	PL	Clock Pulse	OUTPUTS				
			Q_D	Q_C	Q_B	Q_A	TCU

(B) DOWN Counter:

MR	PL	Clock Pulse	OUTPUTS				
			Q _D	Q _C	Q _B	Q _A	TCD

(C - 1) Pre - Settable UP Counter:

Preset Value	Clock Pulse	OUTPUTS			
		Q _D	Q _C	Q _B	Q _A

(C - 2) Pre - Settable DOWN Counter:

Preset Value	Clock Pulse	OUTPUTS			
		Q _D	Q _C	Q _B	Q _A

Result: The working of Pre - Settable Counter using IC 74193 has been studied successfully.

Precautions and Sources of Error:

1. The connections should be made according to the IC pin diagram.
2. **The connections should be tight, otherwise there is a possibility of false counting.**
3. The V_{CC} and ground should be applied carefully at the specified pin only.

QUESTIONS FOR VIVA

Multiplexer & De - Multiplexer

Q. How many control pin will be required for 64:1 MUX?

Q. If square pulse is to be applied as input, what should be max voltage of the square wave?

Encoder & Decoder

Q. What would be the result of cascading an encoder with a decoder?

Q. How many ICs can be selected using 16:4 decoder?

Seven Segment Display

Q. Why we need current limiting resistors prior to 7- segment display.

Q. Compare common anode & common cathode type 7- segment display.

Buffer (Unidirectional & Bidirectional)

Q. Why and where buffers are necessary?

Q. Mention any suitable use of 8-Bit bidirectional buffer?

D - Latch

Q. Where latching must be necessary in communication?

Q. In a 8-Bit latch can we check the IC just by latching one or two bits?

Shift Registers

Q. Counters (Ripple Counter, Decade Counter & Pre -Settable Counter)

Q. If a square wave of frequency 10 KHz is applied to a single JK flip-flop, what will be the time period of the output square wave?

Q. How many JK FF you need to change the frequency of 64 KHz to 4KHz?

3

D/A Converter & A/D Converter

1.1 DIGITAL TO ANALOG CONVERTERS

Digital to analog converter, actually converts a digital binary number into an equivalent analog output voltage signal, proportional to the binary weight of the number. In a 4-bit DAC, the input ranges from $(0000)_B = [0]_H$ to $(1111)_B = [F]_H$, which gets converted into a voltage output ranging from 0 to 5 V.

The most common conversion methods use a binary weighted resistor or R-2R resistor ladder network and a summing amplifier. Both the DAC methods produce a binary weighted sum output; the weights are set by the resistor values. **However, precise & accurate IC based DACs are more reliable than those based on discrete components.**

By connecting multiple inputs to the inverting terminal of the operational amplifier, we can convert it into a summing amplifier circuit. **However, for accurate and error less reading, it is always advised to use resistors with least tolerance & offset null the Op -Amp, otherwise measured voltages will deviate too much from the expected values, specifically for lower bits.**

A summing amplifier in the inverting mode produces the negative sum of any number of input voltages as shown in Fig. 3.1.

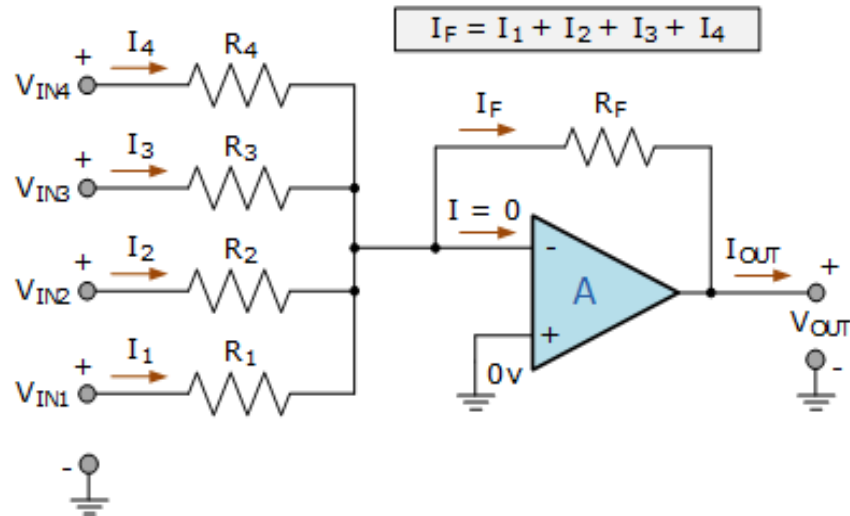


Figure 3.1

In the above summing amplifier circuit, the output voltage, V_{OUT} is proportional to the sum of the four input voltages, V_{IN1} , V_{IN2} , V_{IN3} and V_{IN4} .

$$I_F = I_1 + I_2 + I_3 + I_4 = \frac{V_{IN1}}{R_1} + \frac{V_{IN2}}{R_2} + \frac{V_{IN3}}{R_3} + \frac{V_{IN4}}{R_4} \quad (3.1)$$

$$\therefore V_{OUT} = - \frac{R_F}{R_{IN}} (V_{IN1} + V_{IN2} + V_{IN3} + V_{IN4}) \quad (3.2)$$

If, $R_F = R_1 = R_2 = R_3 = R_4$, then each input channel will have a closed loop voltage gain of unity, so the output voltage is given by

$$V_{OUT} = - (V_{IN1} + V_{IN2} + V_{IN3} + V_{IN4}) \quad (3.3)$$

1.1.2 4-BIT BINARY WEIGHTED DAC

For a 4-bit binary number, there are $2^4 = 16$ possible combinations of A, B, C & D ranging from $(0000)_2$ to $(1111)_2$ which corresponds to decimal 0 to 15 respectively. If we make the weight of each input bit double with respect to the other, we end up with an 8 - 4 - 2 - 1 binary code ratio corresponding to 2^3 , 2^2 , 2^1 and 2^0 .

The resolution of this 8 - 4 - 2 - 1 binary weighted DAC will produce an output voltage change of 0.625 V per bit change in the binary number and is expressed in Table 3.1. The output voltages are all negative due to the inverting mode of the summing amplifier.

Table 3.1

INPUTS				V_{OUT} Expression	V_{OUT}
D	C	B	A	$V_D + \frac{1}{2}V_C + \frac{1}{4}V_B + \frac{1}{8}V_A$	(V)
0	0	0	0	$0*5 + 0*5 + 0*5 + 0*5$	0.00
0	1	1	0	$0*5 + \frac{1}{2}*5 + \frac{1}{4}*5 + 0*5$	- 3.75
1	0	0	0	$1*5 + 0*5 + 0*5 + 0*5$	- 5.00
1	1	0	0	$1*5 + \frac{1}{2}*5 + 0*5 + 0*5$	- 7.50
1	1	1	1	$1*5 + \frac{1}{2}*5 + \frac{1}{4}*5 + \frac{1}{8}*5$	- 9.375

There is a disadvantage here; a binary weighted resistor DAC requires a large range of high precision resistors (one per bit) for an n -bit DAC making it impractical and expensive, for converters with more than a just few bits of resolution. But this

concept can be expanded further by converting it into an R-2R resistor ladder DAC which requires only two precise resistors, namely R and 2R, which can be made more precise by series/parallel combinations of two R's.

Experiment 3.1: To study D/A conversion using Binary Weighted Network.

Apparatus: 5 V TTL supply, dual power supply for Op -Amp, DMM, 10 K Ω pot, resistors (1 K Ω , 2 K Ω , 3.9 K Ω & 8.2 K Ω for binary weight; 1 K Ω for feedback), connecting wires.

Theory: Consider the circuit shown in Fig. 3.2.

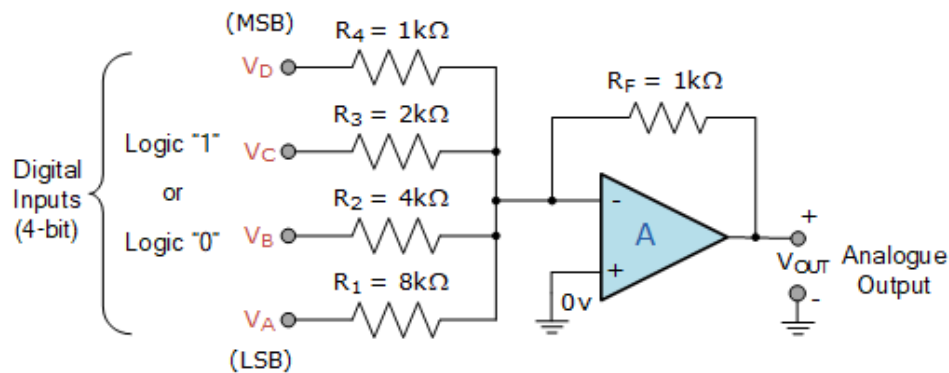


Figure 3.2

Assuming that the four inputs of the summing amplifier are binary inputs with voltage of either 0 or 5 V (LOW or HIGH) and the resistive values of each input resistor is double with regards to the previous one. As a result, an output voltage will be produced which will be the sum of the binary weighted four input voltages and hence, the name 4-bit Binary Weighted D/A Converter.

On labeling the four summing inputs as A, B, C, D and making $R_F = 1\text{ K}\Omega$, with the four input resistors ranging from 1 K Ω to 8 K Ω (or multiples thereof), we can

construct a simple 4-bit binary weighted D/A circuit as shown in Fig. 3.2. The 4-bit binary weighted DAC transfer characteristic are given by

$$V_{\text{OUT}} = - \left[\frac{R_F}{R_4} V_D + \frac{R_F}{R_3} V_C + \frac{R_F}{R_2} V_B + \frac{R_F}{R_1} V_A \right] \quad (3.4)$$

$$V_{\text{OUT}} = - \left[\frac{1 \text{ K}\Omega}{1 \text{ K}\Omega} V_D + \frac{1 \text{ K}\Omega}{1 \text{ K}\Omega} V_C + \frac{1 \text{ K}\Omega}{1 \text{ K}\Omega} V_B + \frac{1 \text{ K}\Omega}{1 \text{ K}\Omega} V_A \right] \quad (3.5)$$

$$V_{\text{OUT}} = - \left[V_D + \frac{1}{2} V_C + \frac{1}{4} V_B + \frac{1}{8} V_A \right] \quad (3.6)$$

Procedure:

- Bread board the circuit as shown in Fig. 3.2.
- Nullify the offset voltage of Op-Amp by suitable method.
- Apply different digital inputs and note the corresponding output voltages in the DMM.
- Compare the output voltages measured with the calculated voltages.

Observations:

DAC Inputs		Analog Outputs	
Inputs	Expected Voltage (V)	Measured Voltage (V)	Error
(0000) _B [00] _D			
(0001) _B [01] _D			
(0011) _B [03] _D			
(0111) _B [07] _D			
(1000) _B [08] _D			
(1010) _B [10] _D			
(1100) _B [12] _D			
(1111) _B [15] _D			

Result: D/A conversion is implemented using binary weighted resistors and the measured analog values are almost tallying with the expected values.

Precautions and Sources of Error:

1. The carbon resistors used should have least tolerance.
2. The offset voltage of Op-Amp must be checked for and if it exists should be nullified.
3. The voltage supplied by all the power supplies used must be verified.

1.1.3 R - 2R BASED DAC

R - 2R DAC is a data converter which uses two precision resistor to convert a digital binary number into an analog output voltage signal proportional to the binary weight of the digital number. Input voltages are applied to the ladder network at various points along its length, more the input points the better resolution of the R-2R ladder. Consider a basic 4-bit R-2R ladder network as shown in Fig. 3.3.

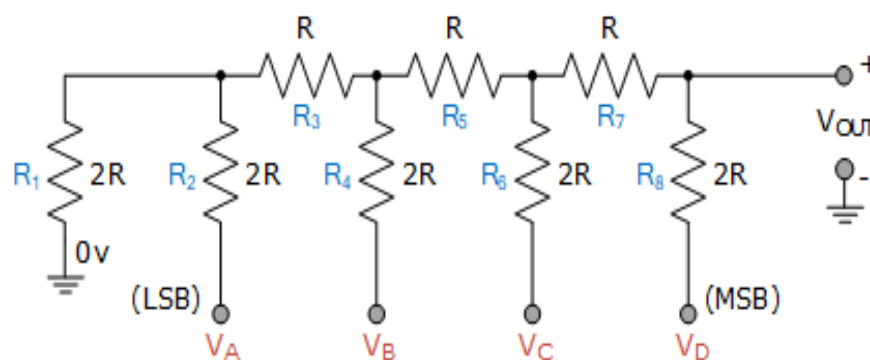


Figure 3.3

Let's assume that all the binary inputs are grounded at 0 V, i.e., $V_A = V_B = V_C = V_D = 0$ V (LOW). The binary code corresponding to these four inputs will therefore be 0000.

Starting from the left hand side and using the simplified equation for two parallel resistors and series resistors, we can find the equivalent resistance of the ladder network as:

$$R_A = R_3 + \frac{R_1 \times R_2}{R_1 + R_2} = R + \frac{2R \times 2R}{2R + 2R} = R + R = 2R$$

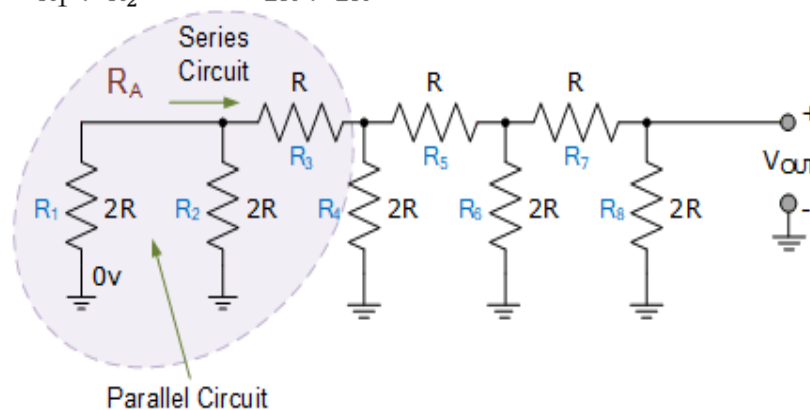


Figure 3.4

Then R_A is equivalent to $2R$. Now we can see that the equivalent resistance R_A is in parallel with R_4 with the parallel combination in series with R_5 . Again we can find the equivalent resistance of this combination and call it R_B .

$$R_B = R_5 + \frac{R_A \times R_4}{R_A + R_4} = R + \frac{2R \times 2R}{2R + 2R} = R + R = 2R$$

So R_B combination is equivalent to $2R$. Similarly, equivalent resistance R_B is in parallel with R_6 with the parallel combination in series with R_7 and so on. Therefore,

with a binary code of 0000 applied as inputs, our basic 4 bit R - 2R digital to analog converter circuit would look like something as shown in Fig. 3.5.

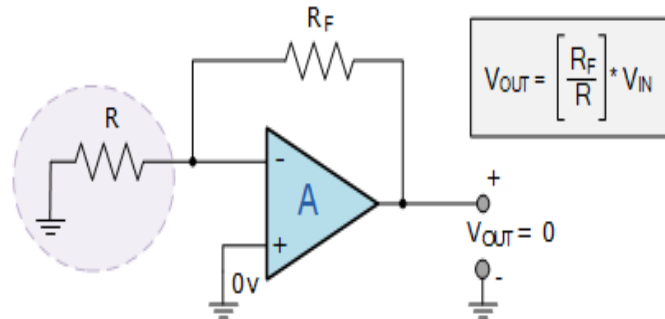


Figure 3.5

So for a 4-bit R - 2R DAC with four grounded inputs (LOW), the output voltage will be zero, thus a 4-bit digital input of 0000 produces an analog output of 0 V.

Fig. 3.6 shows R-2R DAC with input V_A .

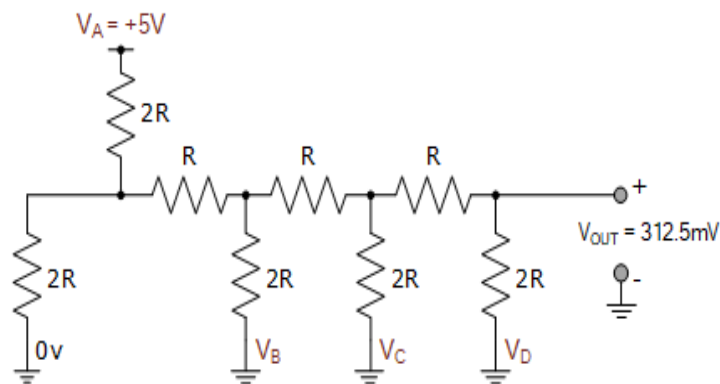


Figure 3.6

Input V_A is HIGH and logic level 1 and all the other inputs are grounded at logic level 0. As the R - 2R ladder network is a linear circuit we can apply network theorem to find equivalent resistance using the same parallel and series resistance calculations as above to calculate the expected output voltage. The output voltage, V_{OUT} is therefore calculated to be 312.5 mV.

Fig. 3.7 shows R -2R DAC with input V_B .

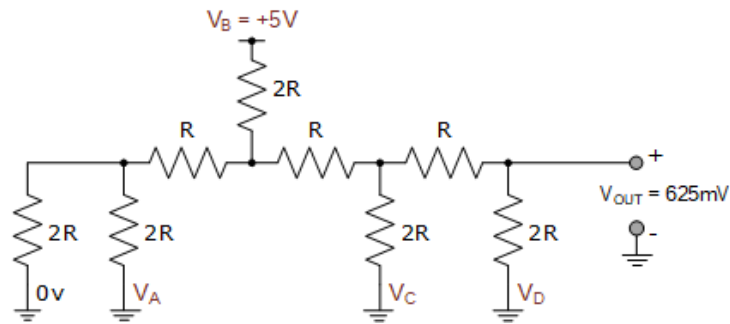


Figure 3.7

Fig. 3.8 shows R -2R DAC with input V_C .

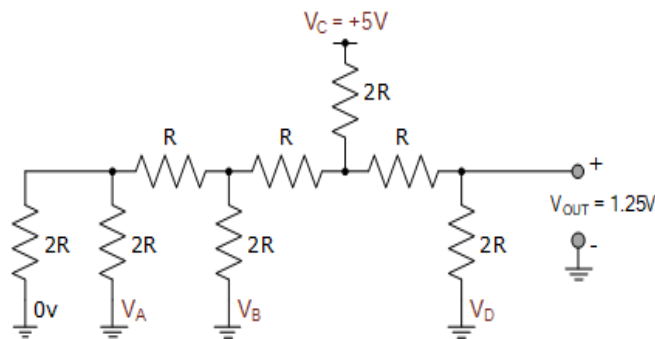


Figure 3.8

Fig. 3.9 shows R - 2R DAC with input V_D .

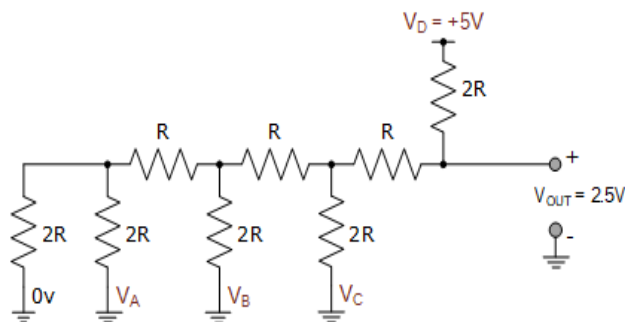


Figure 3.9

From the above results we can see that this voltage is double the output of input bit V_C , 4 times the value of bit V_B and 8 times the value of input bit V_A as it is the 4th bit and therefore classed as the Most Significant Bit, (MSB). Moreover, input V_A represents the LSB and therefore controls the DAC's resolution. The relation for the analog output voltage of 4 - bit D/A converter is

$$V_{OUT} = \frac{V_A + 2V_B + 4V_C + 8V_D}{16} \quad (3.7)$$

where the denominator value of 16 corresponds to the 16 (2^4) possible combinations of inputs to the 4 - bit R - 2R ladder network of the DAC.

The generalized R -2R DAC equation is given by

$$V_{OUT} = \frac{V_A + 2V_B + 4V_C + 8V_D + 16V_E + 32V_F + \dots \text{etc}}{2^n} \quad (3.8)$$

Where n represents the number of digital inputs within the R - 2R resistive ladder network of the DAC producing a resolution of

$$V_{LSB} = \frac{V_{IN}}{2^n} \quad (3.9)$$

Clearly, then the input bit V_A when HIGH will cause the smallest change in the output voltage, while input bit V_D when HIGH will cause the greatest change in the output voltage. Ideally, the ladder network should produce a linear relationship between the input voltages and the analog output as each input will have a step increase equal to the LSB. Table 3.2 shows the expected output voltage values for all 16 combinations of the 4 inputs with + 5 V representing a logic 1 condition.

Table 3.2

INPUTS				V_{OUT} Expression	V_{OUT}
D	C	B	A	$\frac{8 * V_D + 4 * V_C + 2 * V_B + 1 * V_A}{2^4}$	(V)
0	0	0	0	$(0*5 + 0*5 + 0*5 + 0*5)/16$	0.00
0	0	0	1	$(0*5 + 0*5 + 0*5 + 1*5)/16$	0.3125
1	0	0	0	$(8*5 + 0*5 + 0*5 + 0*5)/16$	2.5000
1	0	1	0	$(8*5 + 0*5 + 2*5 + 0*5)/16$	3.1250
1	1	1	1	$8*5 + 4*5 + 2*5 + 1*5$	4.6875

Notice that the full scale analog output voltage for a binary code of 1111 never reaches the same value as the digital input voltage (+ 5 V) but is less by the equivalent of one LSB bit, (3.125 mV in this example). However, the higher the number of digital input bits (resolution) the nearer the analog output voltage reaches full scale when all the input bits are HIGH. Likewise, when all the input bits are LOW, the resulting lower resolution of LSB makes V_{OUT} closer to zero volts.

Experiment 3.2: To study D/A conversion using R -2R Ladder Network.

Apparatus: 5 V TTL supply, dual power supply for Op-Amp, DMM, 10 K Ω pot, resistors (10 K Ω , 20 K Ω ; 1 K Ω for feedback), connecting wires.

Theory: Consider the circuit shown in Fig. 3.10.

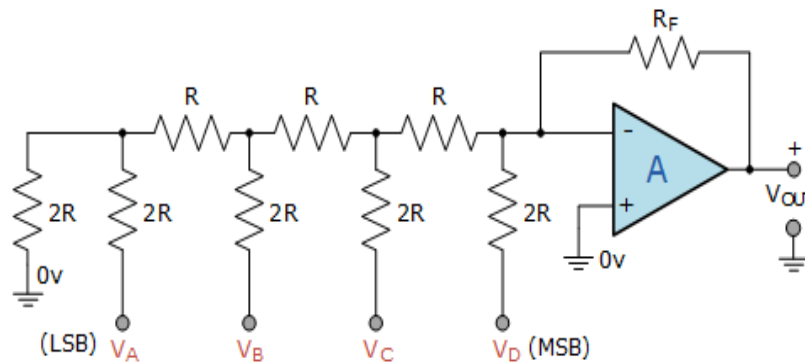


Figure 3.10

The relation for the analog output voltage of 4 - bit D/A converter is

$$V_{\text{OUT}} = \frac{V_A + 2V_B + 4V_C + 8V_D}{16}$$

Where the denominator value of 16 corresponds to the 16 (2^4) possible combinations of inputs to the 4 - bit R - 2R ladder network of the DAC.

Procedure:

- a. Nullify the offset voltage of Op - Amp by suitable method.
- b. Bread board the circuit as shown in Fig. 3.10.
- c. Apply different digital inputs and note the corresponding output voltages in the DMM.
- d. Compare the output voltages measured with the calculated voltages.

Observations:

DAC Inputs	Analog Outputs		
	Expected Voltage (V)	Measured Voltage (V)	Error
(0000) _B [00] _D			
(0001) _B [01] _D			
(0011) _B [03] _D			
(0111) _B [07] _D			
(1000) _B [08] _D			
(1010) _B [10] _D			
(1100) _B [12] _D			
(1111) _B [15] _D			

Result: D/A conversion is implemented using R-2R Ladder network and the measured analog values are almost tallying with the expected values.

Precautions and Sources of Error:

1. The carbon resistors used should have least tolerance.
2. The offset voltage of Op-Amp must be checked for and if it exists should be nullified.
3. The voltage supplied by all the power supplies used must be verified.

Experiment 3.3: To study D/A conversion using 8-Bit IC based DAC.

Apparatus: 5 V TTL supply, dual power supply for Op-Amp, DMM, 10 K Ω pot, two 4.7 K Ω resistors, connecting wires, IC 1408.

Theory: The IC 1408 DAC is an 8-bit R-2R ladder type D/A converter compatible with TTL and CMOS logic. Fig. 3.11 shows the pin diagram for IC 1408 DAC.

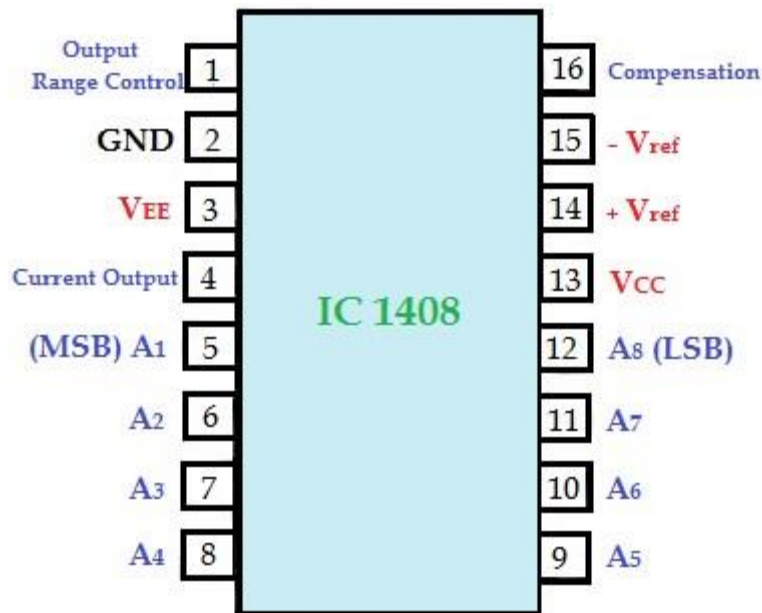


Figure 3.11

The IC 1408 DAC pin diagram consists of a reference current amplifier, an R - 2R ladder and eight high speed current switches. It has eight input data line A₁ (MSB) through A₈ (LSB) which control the positions of current switches.

It requires 2 mA reference current for full scale input and two power supplies V_{CC} = + 5 V and V_{EE} = - 15 V. The voltage V_{ref} and resistor R₁₄ determines the total

reference current source and R_{15} is generally equal to R_{14} to match the input impedance of the current amplifier.

The output current I_O is given by

$$I_O = \frac{V_{ref}}{R_{14}} \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right) \quad (3.10)$$

Fig. 3.12 shows a typical circuit for IC 1408.

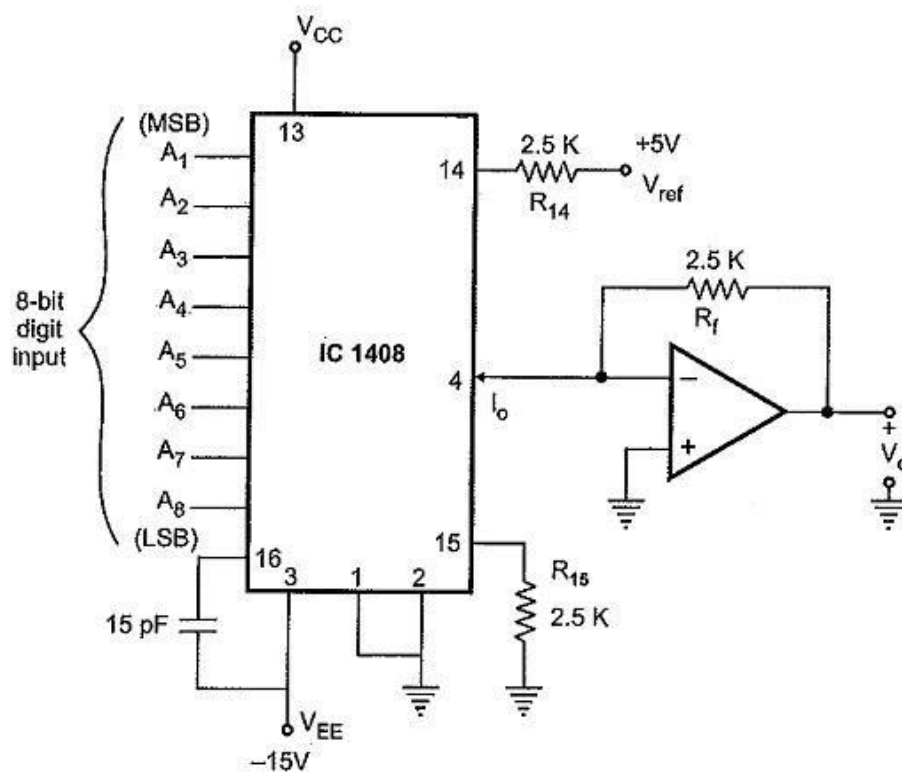


Figure 3.12

Input A_1 through A_8 can be either 0 or 1. Therefore, for a typical circuit full scale current can be given by

$$\begin{aligned} I_O &= \frac{5}{25 \text{ K}} \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} \right) \\ &= \frac{2 \text{ mA} \times 255}{256} = 1.992 \text{ mA} \end{aligned}$$

It shows that the full scale output current is always 1 LSB less than the reference current source of 2 mA. The output voltage for full scale input is given by

$$V_o = 1.992 \times 25 \text{ K} = 4.98 \text{ V}$$

The arrow on the pin 4 shows the output current direction. It is inward. This means that IC 1408 sinks current. At $(0000\ 0000)_2$ binary input it sinks zero current and at $(1111\ 1111)_2$ binary input it sinks 1.992 mA.

The circuit shown in Fig. 3.12 gives the output in the unipolar range. When the digital input is $(00)_{\text{H}}$, the output voltage is 0 V and when the input is $(\text{FF})_{\text{H}}$, the output voltage is +5 V. The circuit can be modified to give bipolar output as shown in Fig. 3.13.

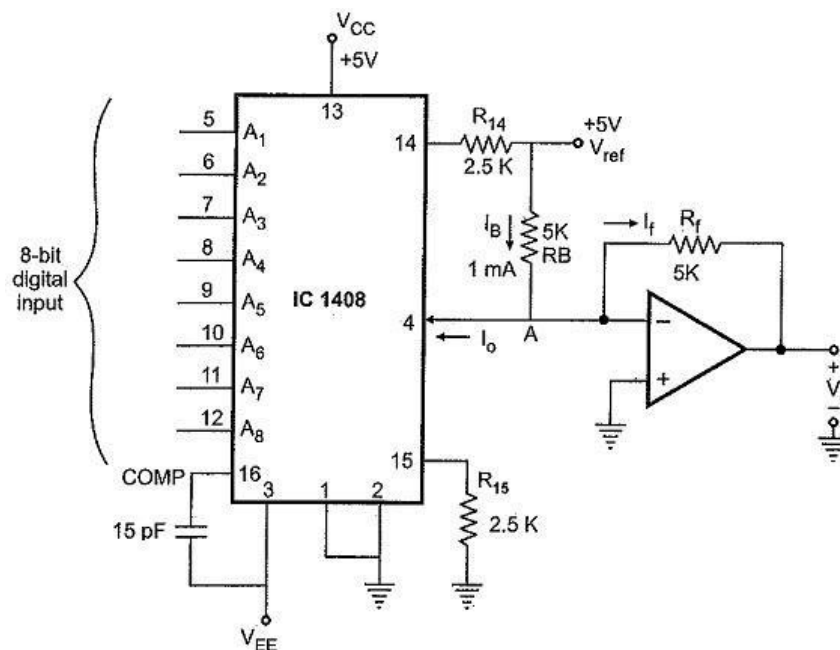


Figure 3.13

Here, resistor R_B (5 K Ω) is connected between V_{ref} and the output terminal of IC 1408 DAC pin diagram. This gives a constant current source of 1 mA.

The circuit operation can be observed for three conditions:

Condition 1: For binary input (00)_H

When binary input is (00)_H, the output current I_O at pin 4 is zero. Due to this current flowing through R_B (1 mA) flows through R_f giving $V_O = -5$ V.

Condition 2: For binary input (80)_H

When binary input is (80)_H, the output current I_O at pin 4 is 1 mA. By applying KCL at node A we get,

$$-I_B + I_O + I_f = 0$$

On substituting values of I_B and I_O , we get

$$I_f = 0$$

and therefore the output voltage is zero.

Condition 3: For binary input (FF)_H

When binary input is (FF)_H, the output current I_O at pin 4 is 2 mA. By applying KCL at node A, we get

$$-I_B + I_O + I_f = 0$$

On substituting values of I_B and I_O , we get

$$I_f = -1 \text{ mA}$$

Therefore, the output voltage is +5 V.

Important Electrical Characteristics for IC 1408:

- Reference current : 2 mA
- Supply voltage : + 5V V_{CC} and - 15V V_{EE}
- Setting time : 300 ns
- Full scale output current : 1.992 mA
- Accuracy : 0.19 %

Procedure:

Nullify the offset voltage of Op-Amp by suitable method.

(A) Unipolar Mode:

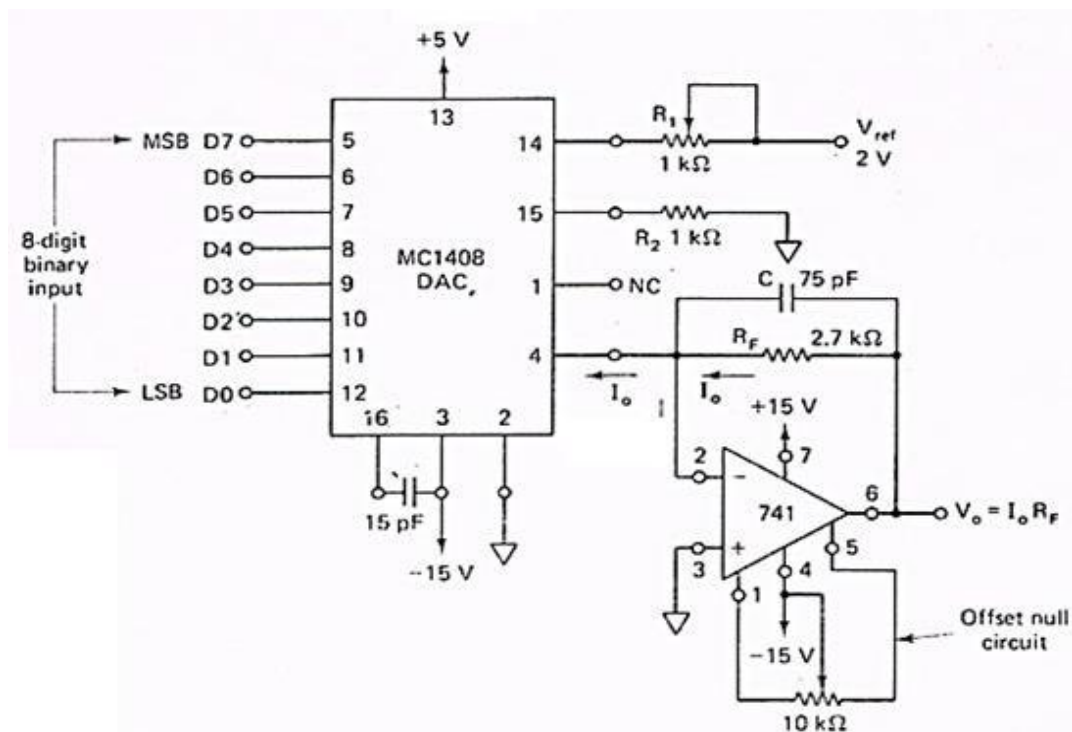


Figure 3.14

- a. Connect the circuit as shown in Fig. 3.14. Adjust the R_{ref} to get reference current 2 mA before connecting the Op-Amp.

- Make all the inputs HIGH and check I_O ; $I_O \approx 2 \text{ mA}$.
- Note corresponding outputs for different input combinations.
- Calculate the output voltage of op - amp by the formula

$$V_O = I_O R_F$$

$$I_O = \frac{V_{\text{ref}}}{R_{14}} \left(\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256} \right)$$

(B) Bipolar Mode:

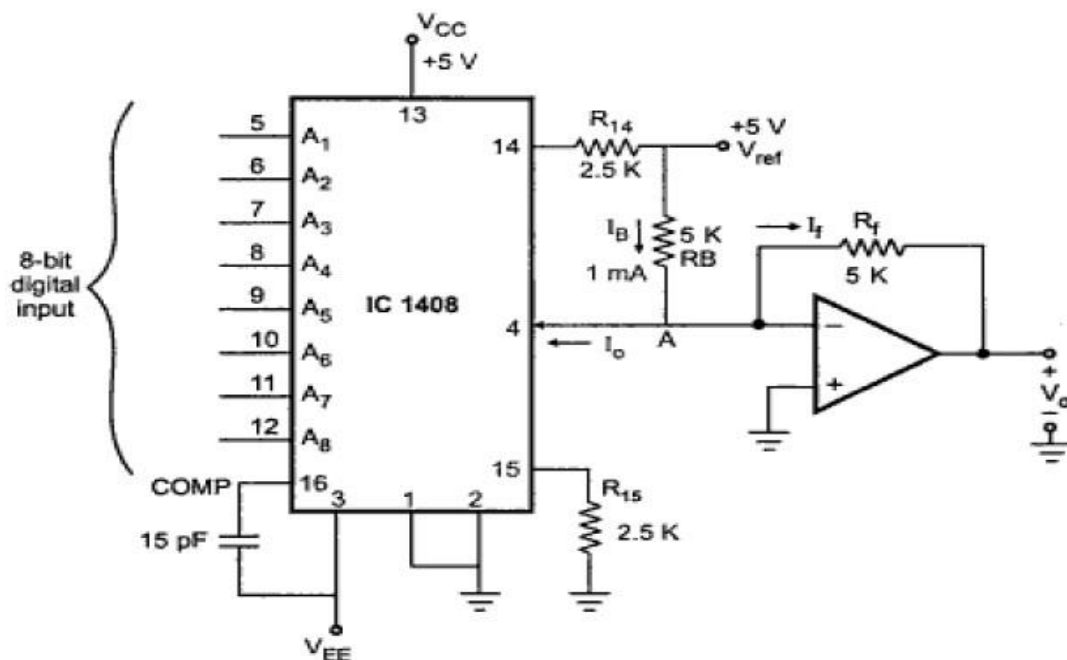


Figure 3.15

- Connect the circuit as shown in Fig. 3.15, by taking $R_B = 2 R_{14}$ and $R_{14} = R_{15}$.
- Adjust the reference current and apply different input combinations, note the corresponding outputs.
- For bipolar mode, output voltage is given by

$$V_O = \frac{V_{ref}}{R_{14}} R_F \left(\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256} \right) - \frac{V_{ref}}{R_B} \times R_F$$

Observations:

(A) Unipolar Mode:

DAC Inputs	Analog Outputs			
	I _O	Expected Voltage (V)	Measured Voltage (V)	Error
(0000 0000) _B [00] _H				
(0000 0001) _B [01] _H				
(0000 0011) _B [03] _H				
(0000 0111) _B [07] _H				
(1000 1000) _B [88] _H				
(1111 1010) _B [FA] _H				
(1100 1100) _B [CC] _H				
(1111 1111) _B [FF] _H				

(B) Bipolar Mode:

DAC Inputs	Analog Outputs			
	I _O	Expected Voltage (V)	Measured Voltage (V)	Error
(0000 0000) _B [00] _H				
(0000 0001) _B [01] _H				
(0000 0011) _B [03] _H				

$(0000\ 0111)_B [07]_H$

$(1000\ 1000)_B [88]_H$

$(1111\ 1010)_B [FA]_H$

$(1100\ 1100)_B [CC]_H$

$(1111\ 1111)_B [FF]_H$

Result: D/A conversion is implemented using 8 - bit DAC IC 1408 and the measured analog values are almost tallying with the expected values.

Precautions and Sources of Error:

1. The carbon resistors used should have least tolerance.
2. The offset voltage of Op-Amp must be checked for and if it exists should be nullified.
3. The voltage supplied by all the power supplies used must be verified.

1.2 ANALOG TO DIGITAL CONVERTERS

Almost every environmental measurable parameter is in analog form like temperature, sound, pressure, light, etc. Consider a temperature monitoring system where in acquiring, analyzing and processing temperature data from sensors is not possible with digital computers and processors. Therefore, this system needs an intermediate device to convert the analog temperature data into digital data in order to communicate with digital processors like microcontrollers and microprocessors.

Analog to Digital Converter (ADC) is an electronic integrated circuit used to convert the analog signals such as voltages to digital or binary form consisting of 1's and 0's. Most of the ADCs take a voltage input as 0 to 10 V, - 5 V to + 5 V, etc., and correspondingly produce digital output as some sort of a binary number.

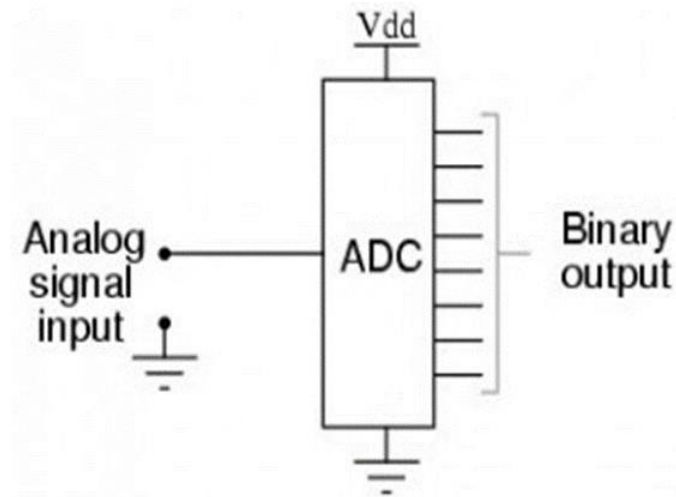


Figure 3.16

There are different types of ADC chips available like the ADC08xx series. A simple ADC can be designed with the help of discrete components. The main features of ADC are sample rate and bit resolution.

- The sample rate of an ADC is nothing but how fast an ADC can convert the signal from analog to digital.
- Bit resolution is nothing but how much accuracy can an analog to digital converter can convert the signal from analog to digital.

Dynamic characteristics of the high performance ADCs are improved measurement repeatability, low power consumption, precise throughput, high linearity, excellent Signal to Noise Ratio (SNR) and so on. A variety of applications of the ADCs are measurement and control systems, industrial instrumentation,

communication systems, and all other sensory based systems. Classification of ADCs based on factors like performance, bit rates, power, cost, etc.

The objective of an A/D converter is to determine the output signal word corresponding to an analog signal. An ADC of 0804 is an 8-bit converter with a 5 V power supply. It can take only one analog signal as input.

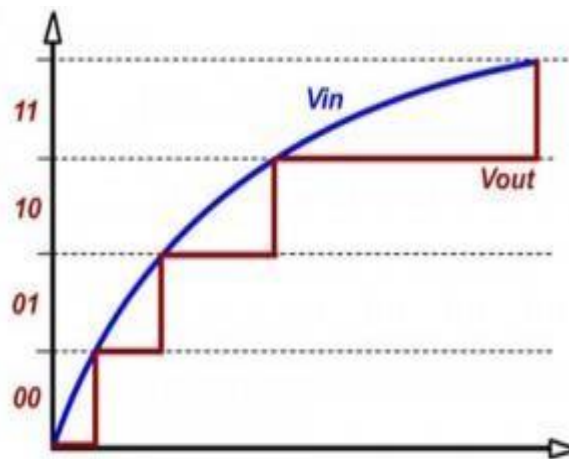


Figure 3.17

Experiment 3.4: To study A/D conversion using 8-Bit IC based ADC.

Apparatus: 5 V TTL supply, DMM, 10 K Ω pot, eight LEDs with eight current limiting resistors of 150 Ω resistors, connecting wires, ADC0804.

Theory: ADC0804 is a very commonly used 8 - bit analog to digital converter. It works with 0 V to 5 V analog input voltage. It has single analog input and 8 - digital outputs. Conversion time is another major factor in judging an ADC, in ADC0804 conversion time varies depending on the clocking signals applied to CLK R and CLK IN pins, but it cannot be faster than 110 μ s.

Fig. 3.18 shows the pin diagram of ADC0804.

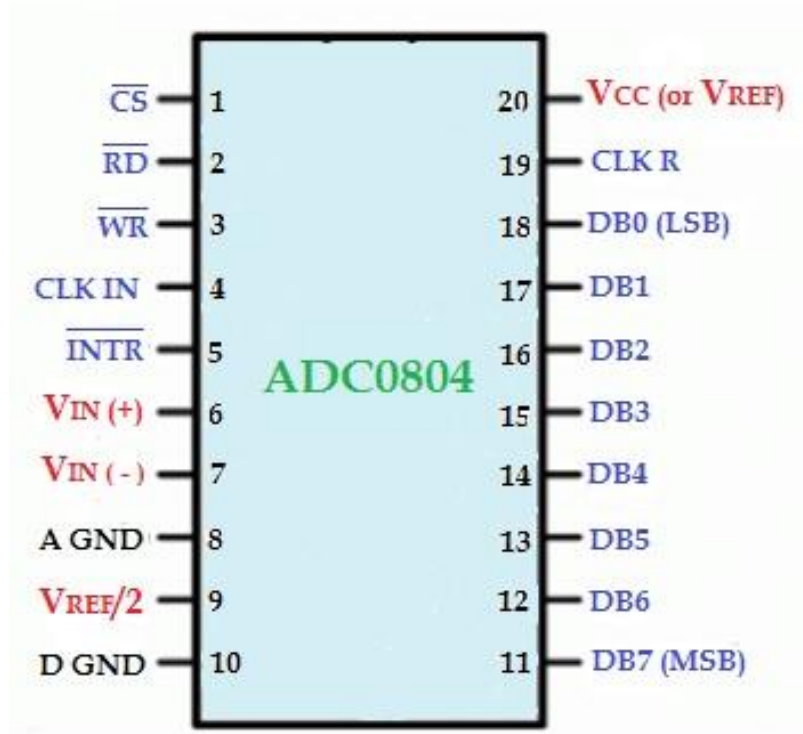


Figure 3.18

Pin Description of ADC0804:

Pin 1: It is a chip select pin and activates ADC, active low.

Pin 2: OE, It is an input pin; high to low pulse brings the data from internal registers to the output pins of ADC after conversion

Pin 3: STC; It is an input pin; low to high pulse is given to start the conversion

Pin 4: It is a clock input pin, to give the external clock

Pin 5: EOC, It is an output pin, goes low when the conversion is complete

Pin 6 & 7: Analog non-inverting input & Analog inverting input.

Pin 8: Ground (0V)

Pin 9: It is an input pin, sets the reference voltage for analog input

Pin 10: Ground (0V)

Pin 11: Pin 18: It is an 8-bit digital output pins

Pin 19: Is used with Clock IN pin when internal clock source is used

Pin 20: Supply voltage; 5V

Features of ADC0804:

- 0V to 5V analog input voltage range.
- Access time is 135 ns.
- Easy interface to all μ Ps & μ Cs.
- Compatible with both MOS and TTL voltage level specifications.
- Works with 2.5V volts as reference.
- Differential analog voltage inputs

It is an 8 - bit converter with a 5 V power supply. It can take only one analog signal as input. The digital output varies from 0 - 255. ADC needs a clock to operate. The time taken to convert the analog to digital value depends on the clock source.

An external clock can be given to CLK IN. Pin 2 is the input pin - High to low pulse brings the data from the internal register to the output pins after conversion. Pin 2 is a Write - Low to high pulse is given to the external clock.

In the standalone configuration, A GND, D GND, IN -, CS, EOC, & OE are all tied to ground. $V_{REF}/2$, by default is at 2.5 V, since it is kept open. The display in the DMM is the analog voltage while the LEDs are the binary/digital form of the analog input.

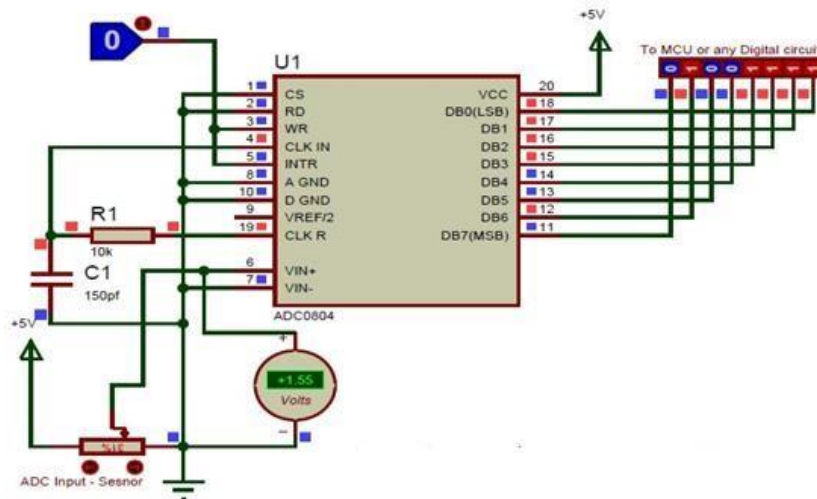


Figure 3.19

Table 3.3

$V_{REF}/2$ (V)	Input Voltage Span (V)	Step Size (V)
Left Open	0 - 5	$5/255 = 19.6$
2	0 - 4	$4/255 = 15.69$
1.5	0 - 3	$3/255 = 11.76$
1.28	0 - 2.56	$2.56/255 = 10.04$
1.0	0 - 2	$2/255 = 7.84$
0.5	0 - 1	$1/255 = 3.92$

Procedure:

- a. Bread board the circuit as shown in Fig. 3.19.
- b. Using a potential divider arrangement or an appropriate sensor, vary the input voltage at V_{IN+} from 0 V to 5 V.
- c. First apply input equal to 0 V and check the digital output at the eight LEDs or the MCU.
- d. Secondly, apply input equal to 5 V and check the digital output at the eight LEDs or the MCU.
- e. Now apply input equal to 2.5 V and check the digital output at the eight LEDs or the MCU.

Observations:

Analog Inputs	Digital Outputs		
	Expected Binary	Measured Binary	Error
0.0 V			
1.0 V			
2.0 V			
3.0 V			
3.5 V			
4.0 V			
4.5 V			
5.0 V			

Result: A/D conversion is implemented using 8-bit ADC IC 0804.

Precautions and Sources of Error:

1. Check voltage across the sensor / potentiometer before connection.
2. Verify that all power supplies are supplying the voltages as expected.
3. Use DMM for accurate results.
4. Use current limiting carbon resistors of 180Ω if output is displayed on LEDs.

QUESTIONS FOR VIVA

4 -Bit Binary Weighted DAC

Q. Mention all the reasons for deviation of observed data from the expected data.

4 -Bit R-2R based DAC

Q. Mention all the reasons for deviation of observed data from the expected data.

Q. How this configuration is better than binary weighted resistor network?

IC based DAC

Q. What is the step size of 8-Bit IC based DAC operating at 5 V?

Q. Why 32-Bit DAC is preferred to 8-Bit DAC. What are its demerits?

IC based ADC

Q. What is the role of SOC, EOC & OE pins in an ADC?

Q. What is the significance of V_{REF} pin in an ADC?

4

μP 8085 Assembly Language Programming:

4.1 INTRODUCTION

The microprocessor is a device consisting of electronic logic circuits manufactured by using either a large scale integration (LSI) or very – large scale integration technique (VLSI). The microprocessor can be divided into three segments for the sake of clarity; arithmetic/logic unit (ALU), register array and control unit.

Features of 8085:

- The 8085 is an 8 – bit general purpose μP and can address 64 K byte of memory.
- 8-bit data bus & 16 – bit address bus.
- A 16-bit program counter and a 16 – bit stack pointer.
- Six 8-bit registers arranged in pairs: BC, DE, HL.
- Requires + 5 V supply to operate at 3.2 MHz single phase clock.
- 4 mask-able and 1 non-mask able interrupt.
- Direct Memory Addressing (DMA) capability.

Fig. 4.1 shows the pin diagram of 8085.



Figure 4.1

The pins on the chip can be broadly grouped into six categories:

- Address Bus

- Data Bus
- Control and Status Signals
- Power Supply and Frequency
- Externally Initiated Signals
- Serial I/O ports

Hardware Model

The hardware model as shown in Fig. 4.2 shows two major segments. One segment includes the arithmetic/logic unit (ALU) and a n 8 - bit register called as accumulator, instruction decoder and flags. The second segment shows 8 - bit and 16 - bit registers. Both segments relate to various internal connections called as internal bus.

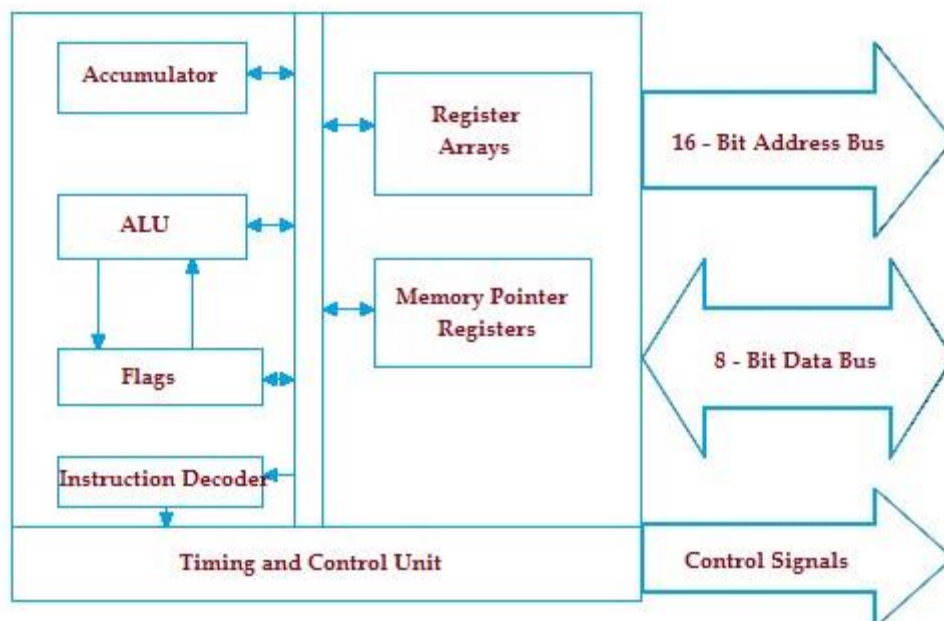


Figure 4.2

4.1.1 8085 PROGRAMMING MODEL

The programming model consists of some segments of the ALU and the registers. This model does not reflect the physical structure of 8085 but includes the information that is critical in writing assembly language programs. The model includes six register, one accumulator and one flag register as shown in Fig. 4.3. In addition it has to 16 - bit registers: the stack pointer and the program counter.

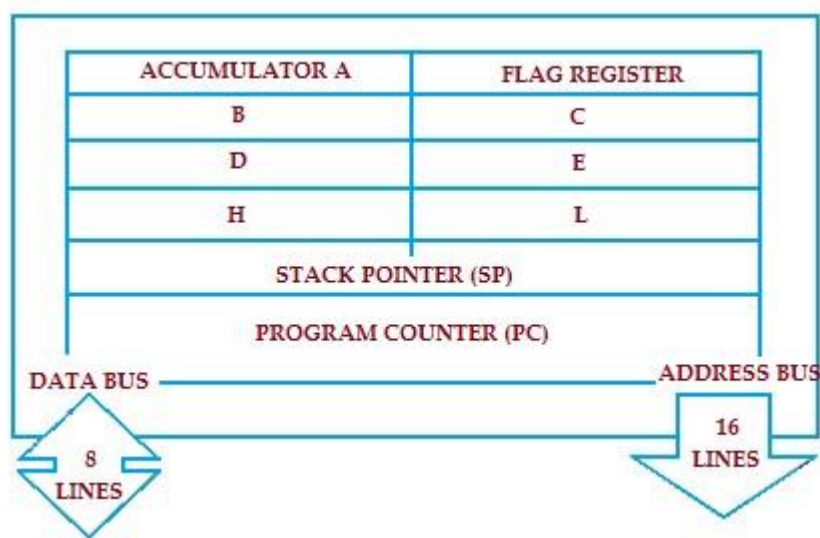


Figure 4.3

Registers: The 8085 has six general - purpose registers to store 8-bit data; these are identified as B, C, D, E, H and L. They can be combined as register pairs BV, DE and HL to perform some 16-bit operations. The programmer can use these registers to store or copy data into registers by using data copy instructions.

Accumulator: The accumulator is an 8 - bit register that is part of the arithmetic/logic unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is also stored in the accumulator. The accumulator is also identified as register A.

Flags: The ALU includes five flip – flops, which are set or reset after an operation according to data conditions of the result in the accumulator and other registers. They are called Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags. The most commonly used flags are Zero, Carry and Sign. The microprocessor uses these to test data conditions.

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

8085 Flags: The following flags are set or reset after the execution of an arithmetic or logical operation; data copy instructions do not affect any flags.

- **Zero (Z):** The Zero flag is set to 1, when the result is zero; otherwise it is reset.
- **Carry (CY):** If an arithmetic operation results in a carry, the CY flag is set; otherwise it is reset.
- **Sign (S):** The sign flag is set if bit D₇ of the result = 1; otherwise it is reset.
- **Parity (P):** If the result has an even number of 1s. the flag is set; for an odd number of 1s, the flag is reset.
- **Auxiliary Carry (AC):** In an arithmetic operation, when a carry is generated by digit D₃ and passed to digit D₄, the AC flag is set. This flag is used internally for BCD (binary coded decimal) operations; there is no Jump instruction associated with this flag.

Program Counter (PC) & Stack Pointer (SP): These are two 16 – bit registers used to hold memory addresses. The size of these registers is 16 – bit because the memory addresses are 16 – bits. The microprocessor uses the PC register to sequence the execution of the instruction. The function of the program counter is to point to the memory address from which the next memory byte is to be fetched. When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location. The stack pointer is also a 16-bit register used as a memory pointer. It points to a memory location in RAM memory called the stack. The beginning of the stack is defined by loading a 16-bit address in the stack pointer.

4.2 μP PROGRAMMING - I

4.2.1 SAMPLE PROGRAM (ADDITION)

MVI B, 25	A	25+AF=D4	F
MVI C, AF	B	25	AF C
MOV A, B	D		E
ADD C	H		L
HLT			

Programs:

***** Drawing flow chart for each program is mandatory**

1. Rewrite the above program for subtraction.

2. Modify the program of addition and subtraction by using minimum number of bytes if result of addition and subtraction is only required.
3. Rewrite the above program for following logical operations:
 - a. AND
 - b. OR
 - c. XOR
4. Modify the above program with minimum number of bytes.
5. Modify the programs of addition and subtraction such that the data is available in certain memory locations and the result should be available in next successive memory location.
6. Write an ALP to add four 8 - bit data. After the execution of the program, the result along with the carry and the data, both should be visible to the programmer through registers.
7. Consider any arbitrary HEX numbers; write an ALP to transfer the number in all registers A, B, C, D, E, H and L. Verify the result after execution.
8. Consider any arbitrary HEX numbers; write an ALP to find its 1's complement and 2's complement, predecessor and successor.

4.3 μP PROGRAMMING - II

4.3.1 MONITOR SUBROUTINES @ DAYNA 8085 KIT

One should be familiar with *Monitor Subroutines* such as *Read Keyboard*, i.e., RDKBD (CD E7 02 located at 02 E7), *Display on Address Field*, i.e., MODIAD (CD 62 03 located at 03 62), *Display on Data Field*, i.e., MODIDT (CD 6E 03 located at 03 6E) and *Delay* (CD F1 05 located at 05 F1) mentioned in the 8085 μP kits manual.

Programs:

- a) Write an ALP to add ten 8 - bit numbers located in consecutive memory locations and display the result in the data field using MS MODIDT.
- b) Consider any two arbitrary HEX numbers, write an ALP to find their Addition, Subtraction, AND, OR, XOR and store the numbers and results in various registers.
- c) Add two 16 - bit HEX numbers using DAD command and use MS MODIAD to display the answer in the address field.
- d) Multiply two 8 - bit HEX numbers using a suitable ALP. Use MS MODIDT to display the answer in the data field.
- e) Write an ALP to find the number of odd numbers from the given set of numbers. Modify the program to find even/positive/negative numbers.

4.4 DELAY

μP processes the data in micro seconds to whom it's slow peripherals cannot respond more over human eye cannot respond to such fast variations.

4.4.1 To generate SMALL Delay

```
                MVI B, 10
LOOP           DCR B
                JNZ     LOOP
                RET
```

4.4.2 To generate BIG Delay

```
                LXI D, FFFF
LOOP    DCX D
MOV A, D

                ORA E
                JNZ     LOOP
                RET
```

4.4.3 To generate BIGGER Delay

```
                MVI B, 10 LOOP I    MVI C, 78
LOOP II  DCR C

                JNZ     LOOP II
                DCR B
                JNZ     LOOP I
                RET
```

Maximum DELAY can be up to 0.293 sec.

4.5 PPI 8255A

The 8255A is a general purpose programmable I/O device designed to transfer the data from simple I/O to interrupt I/O based on the conditions applied. It can be used with almost all Microprocessors & Microcontrollers. It consists of three 8 – bit bidirectional I/O ports (24 I/O pins) can be configured in different modes.

Features of 8085:

- It consists of 3, 8 – bit I/O ports, i.e., Port A, Port B and Port C.
- Address/Data bus must be externally de-multiplexed.
- TTL compatible.
- It has improved DC driving capability.

Ports of 8255A

8255A has three ports, i.e., Port A, Port B and Port C.

- Port A contains one 8 – bit output latch/buffer and one 8 - bit input buffer.
- Port B is similar to Port A.
- Port C can be subdivided into Port C lower (PC₀ – PC₃) and Port C upper (PC₄ – PC₇), using appropriate control word.

These three ports are further divided into two groups, i.e., Group A and Group B. These two groups can be programmed in three different modes, named as Mode 0, Mode 1 and Mode 2.

Fig. 4.4 shows the pin diagram of 8255A.

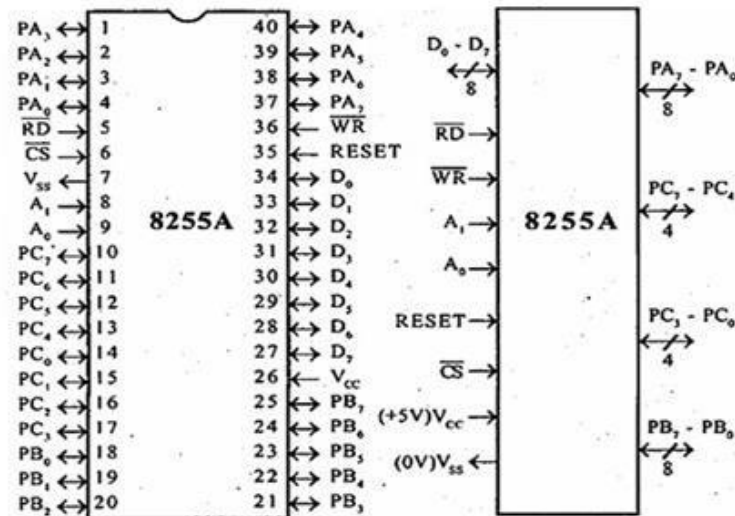


Figure 4.4

Operating Modes

8255A has three different operating modes:

- **Mode 0:** In this mode, Port A and Port B is used as two 8 – bit ports and Port C as two 4 – bit ports, each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.
- **Mode 1:** In this mode, Port A and Port B is used as 8 – bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.
- **Mode 2:** In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.

Fig. 4.5 shows the Control Word format for BSR mode.

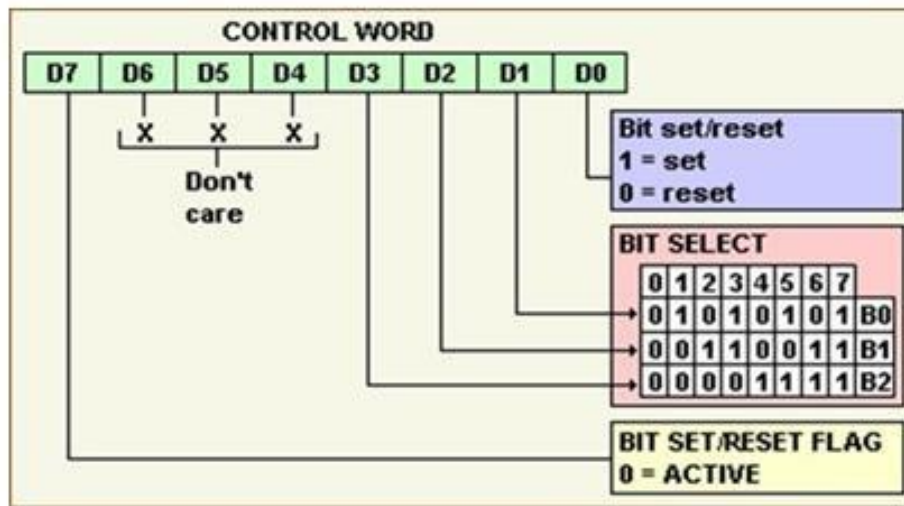


Figure 4.5

Fig. 4.6 shows the Control Word format for I/O mode.

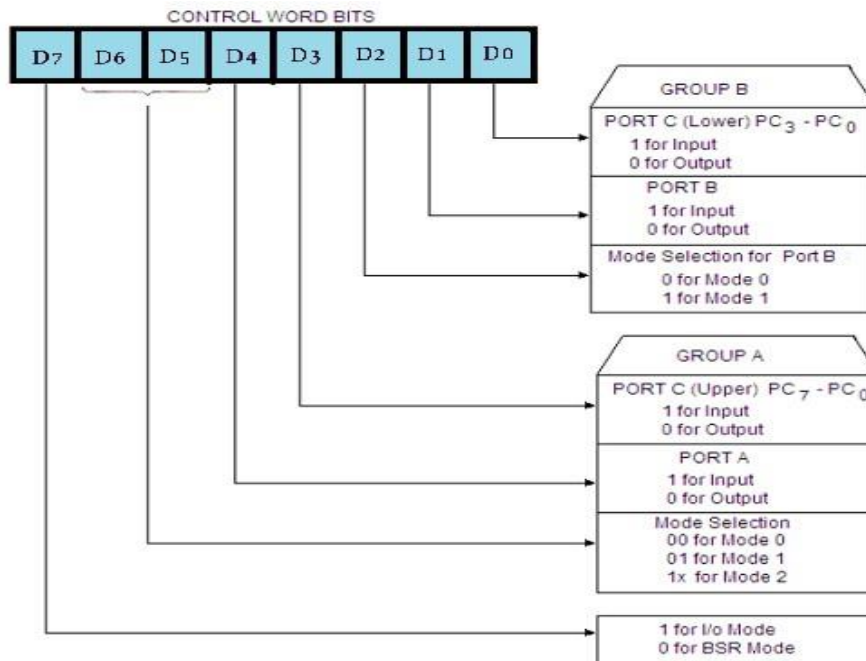


Figure 4.6

Understand Port Interfacing: Write an ALP to input the data from Port A & Port B and display the sum at port C.

Port A	(80)H
Port B	(81)H
Port C	(82)H
Control Register	(83)H

MVI A, 92H	; define & load the control word into the accumulator
OUT 83H	; write the control word the control port of 8255
IN 80H	; data input from port A with address 80H
MOV C, A	; move the input from port A to register C
IN 81H	; data input from port B with address 81H
ADD C	; data from Port A & Port B are added
OUT 82H	; output the result of addition to port C with address 82H

Programs:

***** Drawing flow chart for each program is mandatory**

1. Write an ALP to RESET & SETPC.3 alternatively with an appropriate delay.
2. Write an ALP to blink an LED @ PC.6 with an appropriate delay.
3. Write various ALPs to blink 8 - bit LEDs using μP 8085 and PPI 8255 at appropriate delay.
4. Write an ALP to receive the data @ Port B and Port C and to display the difference at Port A.

5. Write an ALP to receive the data @ Port B and Port A and to display the ORed at Port C.
6. Write an ALP to read the status of the switches and display on the LEDs.
7. Write an ALP so that when the first switch is made ON, all the LEDs should glow and when the second switch is made OFF, all the LEDs should be off.

4.6 WAVEFORM GENERATION

8255A is a popularly used parallel, programmable input - output device. It can be used to transfer data under various conditions from simple input - output to interrupt input - output. It is a general purpose programmable I/O device designed to interface the CPU with its outside world such as ADC, DAC, keyboard, etc.

Addresses:

Port A	(80)H
Port B	(81)H
Port C	(82)H
Control Register	(83)H

*** Port addresses are different for different 8085 kits.

Fig. 4.7 shows the basic diagram for 8-bit DAC.

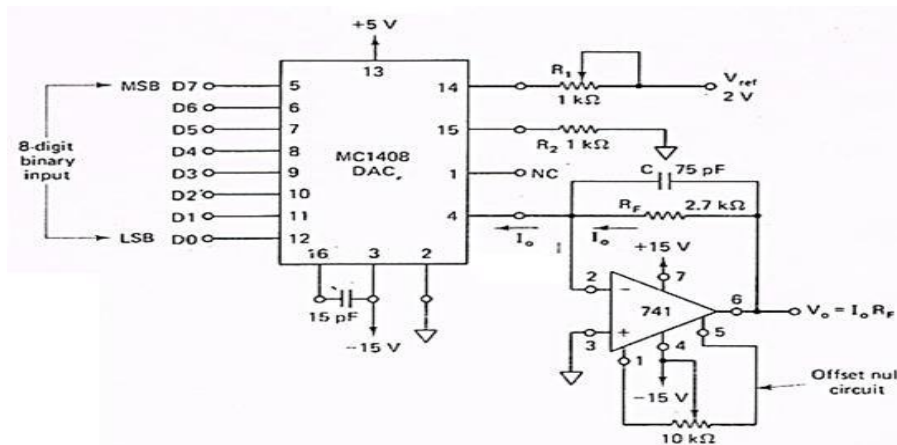
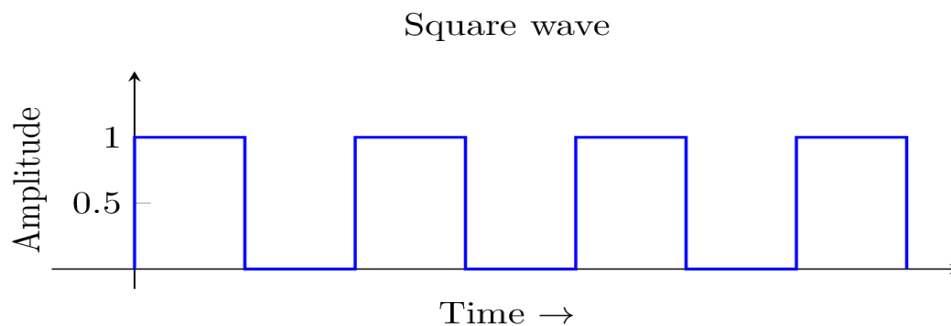


Figure 4.7

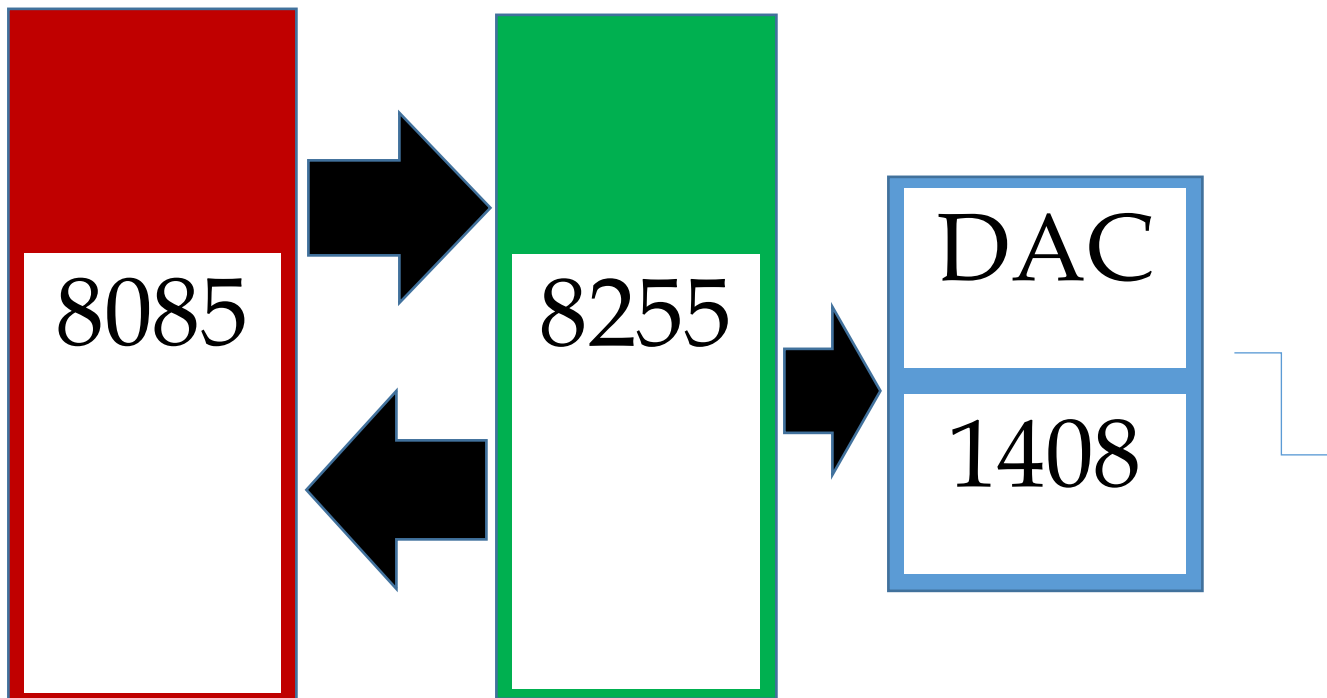
Fig. 4.8 shows a saw tooth waveform. The digital signals coming from μP through 8255 has to be converted into analog signal to see it on CRO. This purpose of conversion is done by a DAC circuit.



ADDRESS	LABEL	OP-CODE	MNEMONICS	COMMENT
C000	START	()H ()H	MVI A (80)H	Initialize Port A as O/P
C002		()H ()H	OUT (83)H	
C004	LOOP 1	()H ()H	MVI A (00)H	Lower value of SW
C006		()H ()H	OUT (80)H	Out at Port A
C008		()H ()H	MVI B, FF	Load register B with count

				FF
C00A	LOOP 2	()H	DCR B	Decrease count of B by 1
C00B		()H ()H	JNZ LOOP 2	JMP till B becomes Zero
C00E		()H ()H	MVI A, FF	Higher value of SW
C010		()H ()H	OUT (80)H	Out at Port A
C012		()H ()H	MVI B, FF	Load register B with count FF
C014	LOOP 3	()H	DCR B	Decrease count of B by 1
C015		()H ()H	JNZ LOOP 3	JMP till B becomes Zero
C018		()H ()H	JMP LOOP 1	To get continuous SW

- Address locations & addresses as per Dyna 8085 kit.



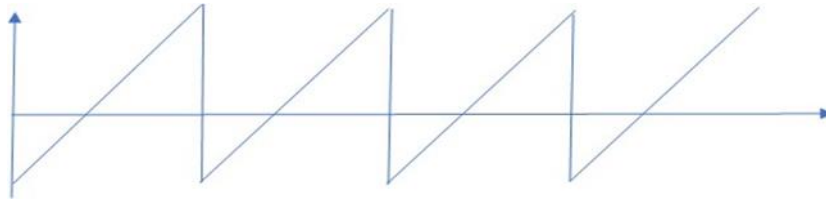


Figure 4.8

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
	START	0H0H	MVIA(80)H	Initialize Port A as O/P
		0H0H	OUT(83)H	
LOOP2		0H0H	MVIA(00)H	Initialize the count
		0H0H	OUT(80)H	Send count to Port A
LOOP1		0H	INR A	Increase content of acc. by one
		0H0H	OUT(80)H	Send count to Port A
		0H0H	CPI (FF)H	Compare the [A] with data (FF)H
		0H0H 0H	JNZLOOP1	Jump if comparison not equal to (FF)H
		0H0H0H	JMP LOOP2	Jump if comparison is equal to (FF)H

Programs:

***** Drawing flow chart for each program is mandatory**

Write an ALP to generate waveforms using μP 8085, PPI 8255 and a suitable DAC.

1. Symmetric/ Asymmetric Square
2. Triangular
3. Ramp Waveform
4. Trapezoid

QUESTIONS FOR VIVA

Basic Concepts

Q. Find the value of $[1111]_2$ in D/H/O number system.

Q. Express the result of addition of 67 and 75 in all the number systems (D/H/O).

Q. Find the binary of $(8453)_{16}$ and $(7421)_8$.

Q. If $(10101010111100001000)_B = (X)_O = (Y)_H$. Find X and Y.

Q. Express the result in HEX of AND, OR and XOR of 4E and FC.

Q. Find: $05 \times 07 = ()_H$.

Q. Determine the 2's complement of $(3C)_H$. What is its significance?

Q. Find the negative of the following numbers: 23, 45, 4D, 5C, 78, 9F, DD, C5.

Q. Find: $34 - D6 = ()$; as expected in the machine.

Q. Find: $D3 + FE = ()$. Also discuss the status of FLAG register.

Q. Find:

a. $4678 + FBCD + 3DEF =$

b. $46FF + F03D + 366F =$

c. $44679 - 3FD2D =$

d. $4DFCD - 3FFFF =$

Q. Find: Even number/Odd numbers/Positive Number/Negative Numbers/Numbers with Odd Parity/Numbers with Even Parity from the set of numbers: 4A, 6B, 8C, 7D, E8, F5, DD and EF.

μP Programming - I

Q. What is special about HL pair as compared to BC and DE pair?

Q. What is the role of carry flag in the process of rotation of 8 - bit data?

Q. Name few instructions which do not affect the Flag Register.

Q. Name few instructions who by pass the Accumulator.

Q. Differentiate between the role of Carry Flag in RLC and RAL.

Q. What is the role of 16 - bit PC?

Q. Which register pair is used for pointing memory location?

Q. For which register pairs the exchange instruction is valid?

Q. What is the role of accumulator in IN/OUT instructions?

PPI 8255A & Waveform Generation

***** Drawing flow chart for each program is mandatory**

Q. Write a Control Word in BSR mode to SETPC.7.

Q. Write a Control Word in BSR mode to RESETPC.0.

Q. Write a Control Word in BSR mode to SETPC.5.

Q. Write a Control Word in I/O mode for all Ports of 8255 as I/P Ports.

Q. Write a Control Word in I/O mode for Port A and Port B as I/P Ports and Port C as O/P Port.

Q. Write a Control Word in I/O mode for all Ports of 8255 as O/P Ports.

Q. Write a Control Word in I/O mode for Port B and Port C as I/P Ports and Port A as O/P Port.

Q. What is the step size of an 8 - bit DAC and a 16 - bit DAC?

5

μC 8051: Assembly Language Programming

5.1 INTRODUCTION

Microcontroller, a computer on a single IC, contains a processor core, ROM, RAM & I/O pins dedicated to performing various tasks. Microcontrollers are generally used in projects and embedded systems. Some examples of μCs are 8051, AVR, PIC, etc.

The most popular microcontroller 8051 belongs to the MCS - 51 family of microcontrollers by Intel. Fig. 5.1 shows the hardware model of microcontroller.

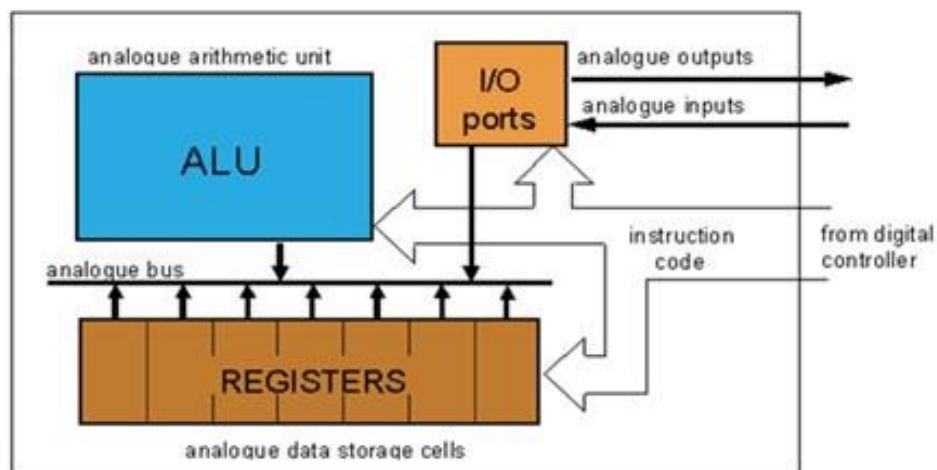
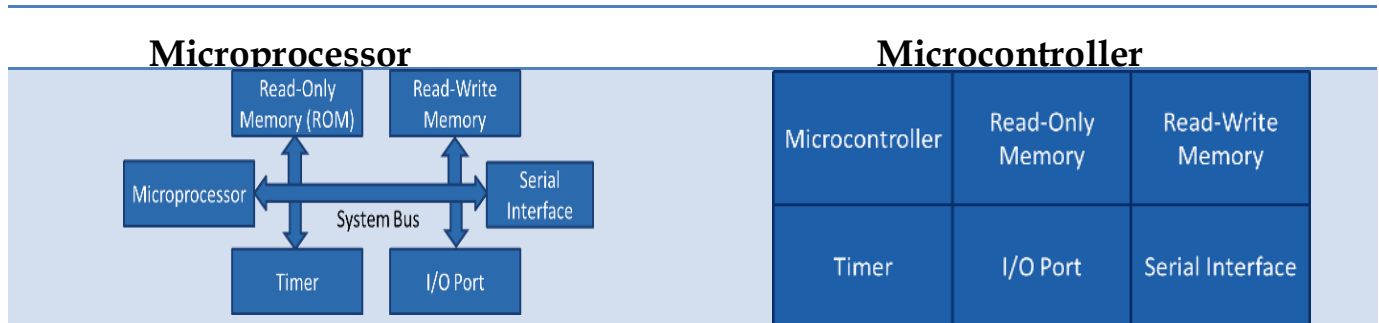


Figure 5.1

Table 5.1 shows a comparative study between microprocessor and microcontroller.

Table 5.1



μP: Heart of Computer System.

μC: Heart of Embedded System.

Memory & I/O are connected externally	μC has processor along with internal memory & I/O components
I/O has to be connected externally; circuit becomes large.	Memory & I/O are present internally, the circuit is compact.
Cannot be used in compact systems & hence inefficient.	Can be used in compact systems & hence efficient
Less cost effective	Cost effective
Due to external components, power consumption is high; not suitable to use with devices running on stored power like batteries.	Since external components are low, total power consumption is less & can be used with devices running on stored power like batteries.
No power saving features.	Idle mode & power saving mode.
Components being external, each instruction will need external operation, hence is relatively slower.	Components are internal hence the operations are internal instruction & so speed is fast.
Smaller number of registers, hence more operations are memory based.	Large number of registers, hence the programs are easier to write.

Based on von Neumann architecture; program & data are stored in same memory module

Based on Harvard architecture where program memory & Data memory are stored separate

Used in personal computers

Used mainly in MP3 players & WM

Fig. 5.2 shows the general schematic diagram of 8051 microcontroller.

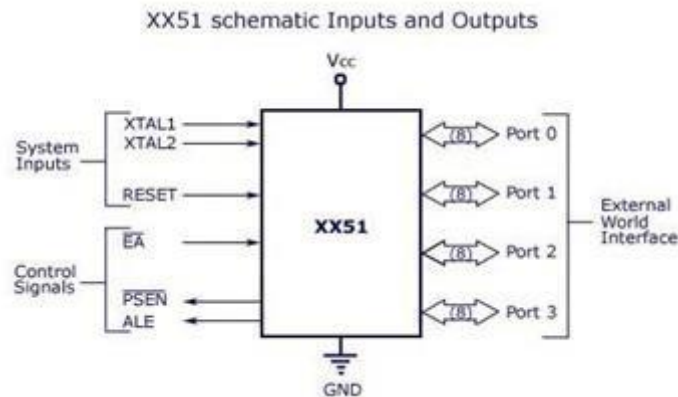


Figure 5.2

There are three system inputs, three control signals and four ports, for external interfacing. V_{CC} supply & ground are with their usual significance. XTAL 1 and XTAL 2 are for the system clock inputs from crystal clock circuit. RESET input is required to initialize microcontroller to default/desired values and to make a new start. There are control signals, EA, PSEN and ALE. These signals known as External Access (EA), Program Store Enable (PSEN) and Address Latch Enable (ALE); are used for external memory interfacing.

If there is no requirement of external memory interfacing then, EA pin is pulled HIGH (connected to V_{CC}); two others PSEN and ALE are left alone. There is $0.1 \mu F$ decoupling capacitor connected to V_{CC} (to avoid HF oscillations at input). There are four ports numbered 0, 1, 2, 3; called as Port 0, Port 1, Port 2 & Port 3 which are used

for external interfacing of devices like DAC, ADC, 7 - segment display, LED, etc. Each port has 8 I/O lines and they all are bit programmable.

Fig. 5.3 shows the pin diagram of 8051.



Figure 5.3

5.1.1 PROGRAM STATUS WORD (PSW) REGISTER

The flag register in the 8051 is called the Program Status Word (PSW) register. Although the PSW register is 8 – bits wide, only six bits are used by 8051. The two unused bits are user definable flags. Four of the flags are called *conditional flags* which imply that they indicate some conditions that results are an instruction is executed. These four flags are **CY (carry)**, **AC (auxiliary carry)**, **P (parity)** and **OV (overflow)**.

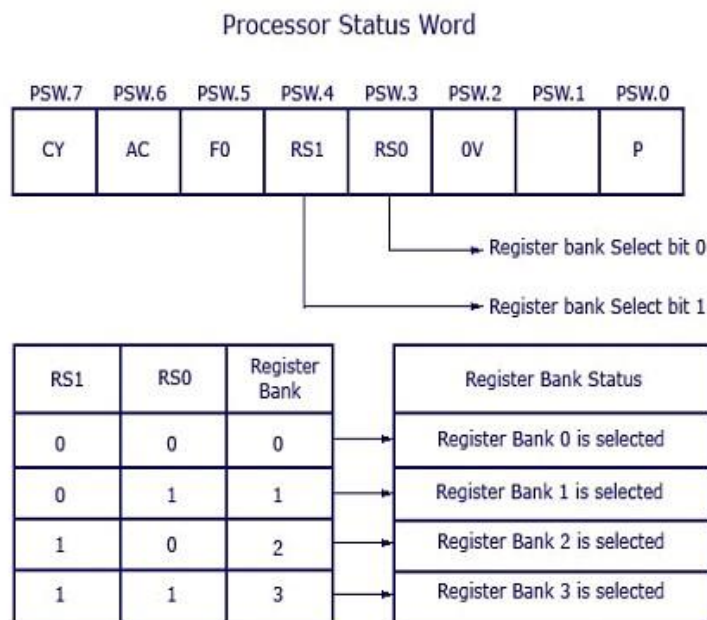


Figure 5.4

From Fig. 5.4, it is clear that the bits PSW.3 and PSW.4 are designated as RS0 and RS1, respectively and are used to change the bank registers. The PSW.5 and PSW.1 bits are general purpose flag bits and can be used by the programmer for any purpose. In other words, they are user – definable.

- **CY (Carry):** This flag is set whenever there is a carry from the D7 bit. This flag bit is affected after an 8 – bit addition or subtraction. It can also be set

to 1 or 0 directly by an instruction such as SETB C and CLR C where SETB C stands for *set bit carry* and CLR C for *clear carry*.

- **AC (Auxiliary Carry):** If there is a carry/borrow from D3 to D4 during an ADD or SUB operation, this bit is set; otherwise it is cleared. This flag is used by instructions that perform BCD (binary coded decimal) arithmetic.
- **P (Parity):** The parity flag reflects the number of 1s in the A (accumulator) register only. If A register contains an odd number of 1's, then P = 1 and P = 0, if A has an even number of 1's.
- **OV (Overflow):** This flag is set whenever the result of a signed number operation is too large, causing the high order bit to overflow into the sign bit. In general, the carry flag is used to find errors in unsigned arithmetic operations. The overflow flag is only used to detect errors in signed arithmetic operations.

5.2 μC PROGRAMMING - I

***** Drawing flow chart for each program is mandatory**

5.2.0 PROGRAM: Addition & storing the Carry in R0

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2000	START	78 00	MOV R0, #00	
2002		75 45 7F	MOV 45, #7F	
2005		74 AF	MOV A, #AF	
2007		25 45	ADD A, 45	
2009		50 01	JNC LOOP1	
200B		08	INC R0	
200C	LOOP1	22	RET	

5.2.1 PROGRAM: Subtraction & storing the Borrow in R0

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2100	START	78 00	MOV R0, #00	
2102		75 45 7F	MOV 45, #7F	
2105		74 AF	MOV A, #AF	
2107		95 45	SUBB A, 45	
2109		50 01	JNC LOOP1	
210B		08	INC R0	
210C	LOOP1	22	RET	

5.2.2 PROGRAM: Multiplication

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2200	START	74 7F	MOVA, #7F	
2202		75 F0 AF	MOV B, #AF	
2205		A4	MUL AB	
2206		22	RET	

5.2.3 PROGRAM: Division

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2300	START	74 7F	MOVA, #7F	
2302		75 F0 AF	MOV B, #AF	
2305		84	DIV AB	
2306		22	RET	

5.2.4 PROGRAM: Data Transfer

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2400	START	7A 0A	MOVR2, #0A	
2402		78 20	MOVR0, #20	
2404		79 30	MOVR1, #30	
2406	LOOP1	E6	MOVA@ R0	
2407		F7	MOV@R1, A	

2408		08	INC R0
2409	09		INC R1
240A		DA FA	DJNZ R2 LOOP1
240C		22	RET

5.2.5 PROGRAM: Addition of Numbers in a Block & Storing the Carry in R1

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2500	START	79 00	MOV R1, #00	
2502		7A 0A	MOVR2, #0A	
2504		78 20	MOV R0, #20	
2506		E6	MOVA, @R0	
2507	LOOP2	08	INC R0	
2508		26	ADD A, @R0	
2509		50 01	JNC LOOP1	
250B		09	INC R1	
250C	LOOP1	DA F9	DJNZ R2 LOOP2	
250E		22	RET	

5.2.6 PROGRAM: Even/Odd Numbers and storing them

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2600	START	78 20	MOV R0, # 20	
2602		79 30	MOV R1, #30	
2604		7A 00	MOV R2, #00	
2606		7B 0A	MOV R3, #0A	
2608	LOOP1	E6	MOV A, @R0	
2609		13	RRC A	
260A		40 04	JC LOOP2	
260C		33	RLC A	
260D		F7	MOV @R1, A	
260E		09	INC R1	
260F		0A	INC R2	
2610	LOOP2	08	INC R0	
2611		DB F5	DJNZ R3 LOOP1	
2613		22	RET	

5.2.7 PROGRAM: Positive/Negative Numbers and storing them

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2700	START	78 20	MOV R0, #20	
2702		79 30	MOV R1, #30	
2704		7A 00	MOV R2, #00	
2706		7B 0A	MOV R3, #0A	
2708	LOOP2	E6	MOV A, @R0	
2709		33	RLC A	
270A		40 04	JC LOOP 1	
270C		13	RRC A	
270D		F7	MOV @R1, A	
270E		0A	INC R2	
270F		09	INC R1	
2710	LOOP1	08	INC R0	
2711		DB F5	DJNZ R3 LOOP2	
2713		22	RET	

5.2.8 PROGRAM: Maximum Number from a set of Numbers

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2800	START	78 20	MOV R0, #20	
2802		79 04	MOV R1, #04	
2804		E6	MOV A, @R0	
2805	LOOP2	08	INC R0	
2806		FA	MOV R2, A	
2807		96	SUBB A, @R0	
2808		50 03	JNC LOOP1	
280A		E6	MOV A, @R0	
280B		80 01	SJMP LOOP3	
280D	LOOP1	EA	MOV A, R2	
280E	LOOP3	D9 F5	DJNZ R1 LOOP2	
2810		08	INC R0	
2811		F6	MOV @R0, A	
2812		22	RET	

5.2.9 PROGRAM: Arranging Numbers in Descending Order

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2900	START	7A 04	MOVR2, #04	
2902	LOOP4	7820	MOVR0, #20	
2904		79 04	MOVR1, #04	
2906	LOOP3	E6	MOV A, @R0	
2907		08	INC R0	
2908		FB	MOV R3, A	
2909		96	SUBB A, @R0	
290A		50 08	JNC LOOP1	
290C		E6	MOV A, @R0	
290D		18	DEC R0	
290E		F6	MOV @R0, A	
290F		08	INC R0	
2910		EB	MOVA, R3	
2911		F6	MOV @R0, A	
2912		80 04	SJMP LOOP2	
2914	LOOP1	EB	MOVA, R3	
2915		18	DEC R0	
2916		F6	MOV @R0, A	
2917		08	INC R0	

2918	LOOP2	D9EC	DJNZ R1 LOOP3
2919		DAE6	DJNZ R2 LOOP4
291B		22	RET

5.2.10 PROGRAM: Cube of number 1 to 6 and storing them in consecutive memory locations

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
6100	START	78 06	MOV R0, #06H	
6102		79 30	MOV R1, #30H	
6104	LOOP1	E7	\$MOVA,@R1	
6105		F5 F0	MOV F0, A	
6107		A4	MUL AB	
6108		87 F0	MOV F0, @R1	
610A		A4	MUL AB	
610B		F7	MOV @R1, A	
610C		09	INC R1	
610D		D9 F5	DJNZ R0 LOOP1	
610F		22	RET	

5.2.11 PROGRAM: Rotation of 32 - bit Register

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
6200	START	74 00	MOV A, #00H	
6202	LOOP1	7D 02	MOV R5, #02H	
6204		12 06 1D	LCALL CLRF	
6207		FB	MOV R3, A	
6208		12 05 9E	LCALL NOUT	
620B		12 01 14	LCALL DELAY	
620E		04	INC A	
620F		80 F1	SJMP LOOP1	

5.2.12 PROGRAM: Counting and Displaying HEX numbers starting from 00H.

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
6000	START	E8	MOV A, R0	
6001		33	RLC A	
6002		F8	MOV R0, A	
6003		E9	MOV A, R1	
6004		33	RLC A	
6005		F9	MOV R1, A	
6006		EA	MOV A, R2	

6007		33	RLC A	
6008		FA	MOV R2, A	
6009		EB	MOV A, R3	
600A		33	RLC A	
600B		FB	MOV R3, A	
600C		E8	MOV A, R0	
600D		92 E0	MOV E0, C	
600F		F8	MOV R0, A	
6010		22	RET	

5.3 μC PROGRAMMING - II THROUGH DPTR

***** Drawing flow chart for each program is mandatory**

5.3.0 PROGRAM: Addition

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2000	START	90 20 40	MOV DPTR, #2040	
2003		E0	MOVX A, @DPTR	
2004		F8	MOV R0, A	
2005		A3	INC DPTR	
2006		E0	MOVX A, @DPTR	

2007		28	ADD A, R0
2008		A3	INC DPTR
2009		F0	MOVX @DPTR, A
200A		22	RET

5.3.1 PROGRAM: Subtraction

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2100	START	90 20 40	MOV DPTR, #2040	
2103		E0	MOVX A, @DPTR	
2104		F8	MOV R0, A	
2105		A3	INC DPTR	
2106		E0	MOVX A, @DPTR	
2107		C3	CLR C	
2108		98	SUBB A, R0	
2109		A3	INC DPTR	
210A		F0	MOVX @DPTR, A	
210B		22	RET	

5.3.2 PROGRAM: Multiplication

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2200	START	90 20 40	MOV DPTR, #2040	
2203		E0	MOVX A, @DPTR	
2204		F5 F0	MOV B, A	
2206		A3	INC DPTR	
2207		E0	MOVX A, @DPTR	
2208		A4	MUL AB	
2209		A3	INC DPTR	
220A		F0	MOVX @DPTR, A	
220B		A3	INC DPTR	
220C		E5 F0	MOVA, B	
220E		F0	MOVX @DPTR, A	
220F		22	RET	

5.3.3 PROGRAM: Division

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2300	START	90 20 40	MOV DPTR, #2040	
2303		E0	MOVX A, @DPTR	
2304		F5 F0	MOV B, A	

2306		A3	INC DPTR	
2307		E0	MOV A, @DPTR	
2308		84	DIV A, B	
2309		A3	INC DPTR	
230A		F0	MOVX @DPTR, A	
230B		A3	INC DPTR	
230C		E5F0	MOV A, B	
230E		F0	MOVX @DPTR, A	
230F		22	RET	

5.3.4 PROGRAM: Square of the Numbers

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2400	START	78 06	MOV R0, #06	
2402		90 20 40	MOV DPTR, #2040	
2405	LOOP1	E0	MOV A, @DPTR	
2406		F5 F0	MOV B, A	
2408		A4	MUL AB	
2409		F0	MOV @DPTR, A	
240A		A3	INC DPTR	
240B		D8 F8	DJNZ R0 LOOP1	
240D		22	RET	

5.3.5 PROGRAM: Cube of the Numbers

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2500	START	90 20 40	MOV DPTR, #2040	
2503		78 06	MOV R0, #06	
2505	LOOP1	E0	MOVX A, @DPTR	
2506		F5 F0	MOV B, A	
2508		A4	MUL A B	
2509		F5 F0	MOV B, A	
250B		E0	MOV A, @DPTR	
250C		A4	MUL A B	
250D		F0	MOV @DPTR, A	
250E		A3	INC DPTR	
250F		D8 F4	DJNZ R0 LOOP1	
2511		22	RET	

5.3.6 PROGRAM: Even Number

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2600	START	78 0A	MOV R0, #0A	
2602		79 00	MOV R1, #00	
2604		90 20 40	MOV DPTR, #2040	
2607	LOOP1	E0	MOV A, @DPTR	

2608		13	RRC A	
2609		4001	JC LOOP2	
260B		09	INC R1	
260C	LOOP2	A3	INC DPTR	
260D		D8F8	DJNZ R0 LOOP1	
260F		E9	MOV A, R1	
2610		F0	MOV @DPTR, A	
2611		22	RET	

Programs:

***** Drawing flow chart for each program is mandatory**

- Write an ALP to transfer content of source memory area that begins from location (XX)_H to the destination memory area that begins from location (YY)_H.
- Write an ALP to find the minimum number from an array of 8-bit numbers stored from (XX)_H to (YY)_H and store it at the memory location(ZZ)_H.
- Write an ALP to find the maximum number from an array of 8-bit numbers stored from (XX)_H to (YY)_H and store it at the memory location(ZZ)_H.
- Write an ALP to find the number of Even/Odd numbers from an array of 8-bit numbers stored from (XX)_H to (YY)_H and store it at the memory

location(ZZ)_H.

- e) Write an ALP to count and display decimals numbers from 0 to 99.
- f) Write an ALP to find the largest number in each of the two memory blocks and multiply them
 - i. Block 1 starts from H and ends at H.
 - ii. Block 2 starts from H and ends at H.
 - iii. Store the result at H.

5.4 DELAY

Microcontroller processes the data in micro seconds to whom it's slow peripherals cannot respond, moreover, human eye also cannot respond to very fast variations.

5.4.1 Generate SMALL Delay

```
{
TOGGLE: MOV P1, #01           ; move 00000001 to the port P1
CALL  DELAY                  ; execute the delay
MOV A, P1                    ; move P1 value to the accumulator
CPL A                        ; complement A value
MOV P1, A                    ; move 11111110 to the port1 register CALL
DELAY                        ; execute the delay
SJMP TOGGLE }

DELAY:  MOV R5, #10H         ; load register R5 with 10
TWO:    MOV R6, #200        ; load register R6 with 200
```

```
ONE:   MOV R7, #200      ; load register R7 with 200
$ DJNZ R7                ; decrement R7 till it is zero
```

```
DJNZ R6                ; decrement R6 till it is zero
```

```
DJNZ R5                ; decrement R5 till it is zero
```

500μS time delay in μC 8051

```
MOV TMOD, #10H        ; select the timer mode by the registers
```

```
MOV TH1, #0FEH        ; store the delay time in TH1
```

```
MOV TL1, #32H         ; store the delay time in TL1
```

```
SETB TR1              ; to start the Timer
```

```
$ JNB TF1 $           ; decrement the value of the timer till it is
zero
```

```
CLR TF1               ; clear the timer flag bit to stop the timer
```

Programs:

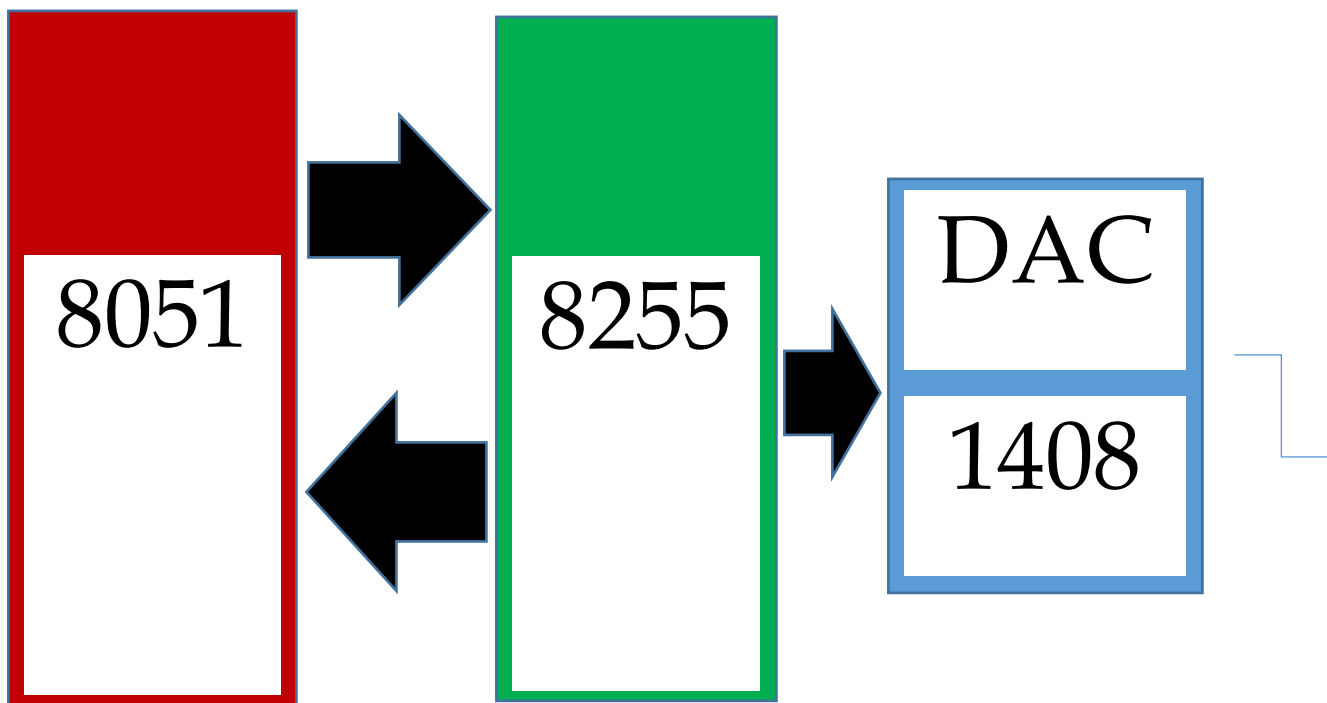
***** Drawing flow chart for each program is mandatory**

- Write an ALP by using appropriate monitor routine to implement flashing display of message "PHYSICS" with suitable delay.
- Write an ALP to implement hex counter that starts counting from (00)_H to (FF)_H with suitable delay. The count should be displayed on 7-segment output console of the kit by using appropriate monitor subroutine.

- c) Write an ALP to implement decimal counter that starts counting from (00)_D to (99)_D with suitable delay. The count should be displayed on 7-segment output console of the kit, by using appropriate monitor subroutine.
- d) Write an ALP for:
- i. Flashing of LEDs.
 - ii. Running light effect on LEDs.
 - iii. Binary up counter display on LEDs.
- e) Four switches A, B, C, D are connected to P1.7, P1.6, P1.5 and P1.4 respectively and a relay driver at P1.0. Write an ALP that monitors these switches and will trip the relay only when the following conditions are satisfied:
- $$Y = A B C D + A B \bar{C} D + A B C \bar{D} + A B C D$$
- f) Write an ALP that will monitor 4 switches connected to (P1.7, P1.6, P1.5, P1.4) and drive 4 LEDs connected to (P1.3, P1.2, P1.1, P1.0) such that:
- i. LED1 (P1.3) will be ON if anyone switch is closed.
 - ii. LED2 (P1.2) will be ON if any two switches are closed.
 - iii. LED3 (P1.1) will be ON if any three switches are closed.
 - iv. LED4 (P1.0) will be ON if all the switches are closed.
- g) Write an ALP to monitor a light beam with the light sensor circuit and an appropriate buffer, connected to pin P1.0, the display should show the count in hexadecimal, which indicates the number of interruptions that takes place. Modify the program for decimal counts.
- h) Write a main program to display the message "ELECT1" on the display of the kit using appropriate sub-routine. Write an ISR program to display the

message “PSWALU” on the display of the kit when an interrupt occurs. Use external interrupt INT0/INT1. (Refer kit manual to find the addresses of various utility program).

5.5 WAVEFORM GENERATION



Program: Square Wave Generation

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENT
2700	START	74 80	MOV A, #80	
2702		90 FF 03	MOV DPTR, #FF03	
2705		F0	MOV @DPTR, A	

2706	LOOP1	90 FF 00	MOV DPTR, #FF00
2709		74 FF	MOV A, #FF
270B		F0	MOV @DPTR, A
270C		79 FF	MOV R1, #FF
270E	LOOP2	7A FF	MOV R2#FF
2710	LOOP3	DA FE	DJNZ R2 LOOP3
2712		D9 FA	DJNZ R1 LOOP2
2714		74 00	MOV A, #00
2716		F0	MOV @DPTR, A
2717		79 FF	MOV R1, #FF
2719	LOOP4	7A FF	MOV R2, #FF
271B	LOOP5	DA FE	DJNZ R2 LOOP5
271D		D9 FA	DJNZ R1 LOOP4
271F		80 E5	SJMP LOOP1

Programs:

***** Drawing flow chart for each program is mandatory**

1. Write an ALP to generate Rectangular, Triangular & Saw tooth waveforms using μC 8051, PPI 8255 and a suitable DAC
2. Write an ALP for μC 8051 & PPI 8255 interface, in I/O mode to receive the data at Port A and Port C and to display the addition at Port B.

μP 8051 Assembly Language Programming:

3. Write an ALP for μC 8051 & PPI 8255 interface, in I/O mode to receive the data at Port A and Port B and to display the ANDed data at Port C.
4. Write various ALPs to glow/blink LEDs representing 8 - bit LEDs using μC 8051, PPI 8255 and an 8 - bit DAC.

Appendix - A

BCD to 7 - Segment Display Decoder

The 74LS47 display decoder receives the BCD code and generates the necessary signals to activate the appropriate LED of 7-segment display, corresponding to the number of pulses applied.

74LS47 decoder is designed for driving a common anode display, a LOW (logic - 0) output will illuminate an LED segment while a HIGH (logic - 1) output will turn it OFF.

For normal operation, the \overline{LT} (lamp Test), \overline{BI} \overline{RBO} (Blanking Input/Ripple Blanking Output) and \overline{RBI} (Ripple Blanking Input) must all be open or connected to logic - 1 (HIGH).

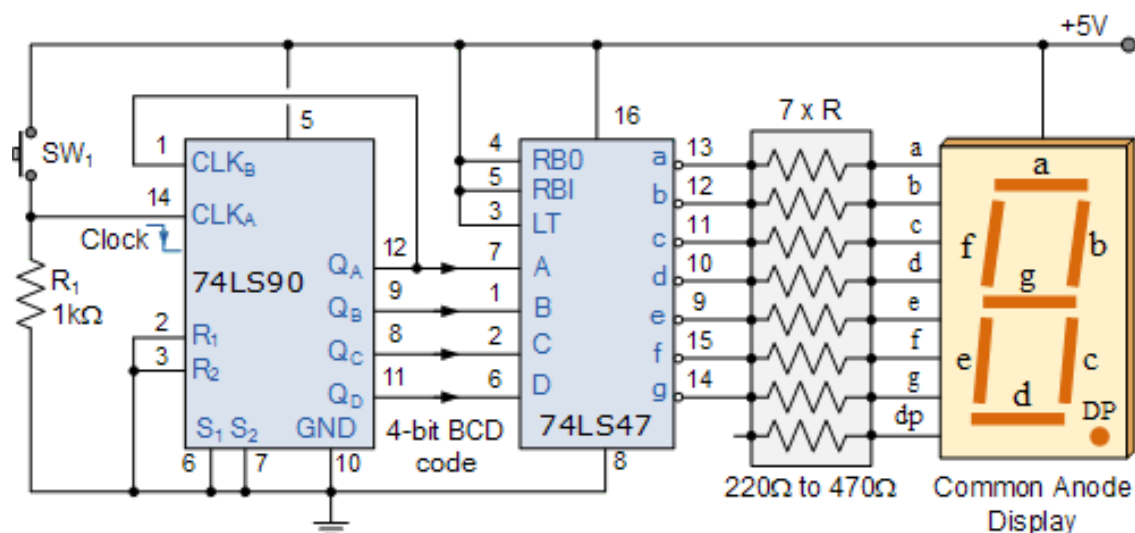


Figure A.1

The 74LS47 binary coded decimal inputs can be connected to the corresponding outputs of the 74LS90 BCD Counter to display the count sequence on the 7 - segment display as shown in Fig. A.1 each time the push button SW1 is pressed. By changing the position of the push button and 1 K Ω resistor, the count can be made to change on the activation or release of the push button switch, SW1.

A 7 - segment display is made of seven individual light emitting diodes to form the display. The best method of limiting the current through a seven segment display is to use a current limiting resistor in series with each of the seven LEDs as shown in Fig. A.2.

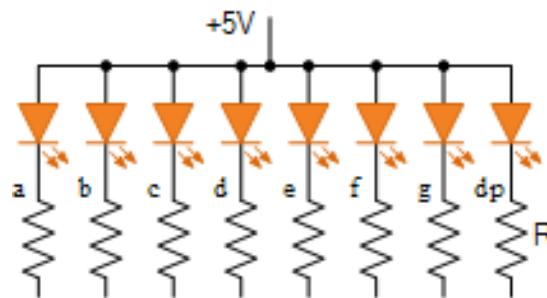


Figure A.2

The advantage here is that the brightness of a particular LED segment does not depend on the state of the other six LEDs, giving the display a constant brightness. The values of the current limiting resistors can be chosen to provide the correct amount of brightness as the amount of ambient light will also determine the required LED intensity.

Fig. A.1 shows a simple 0 to 9 counter using a **7LS90 BCD Counter** and a 74LS47, 7 - segment display driver. To count above 10 and produce a 2 - digit base - 10 counter and display, we need to cascade two separate divide - by - ten counters

together. A 2 - digit BCD counter would count in decimal from 00 to 99 and then reset back to 00.

Likewise if we want to count from 0 up to 999, then three cascaded decade counters are required. Multiple decade counters can be constructed simply by cascading together individual BCD counter circuits, one for each decade as shown in Fig. A.3.

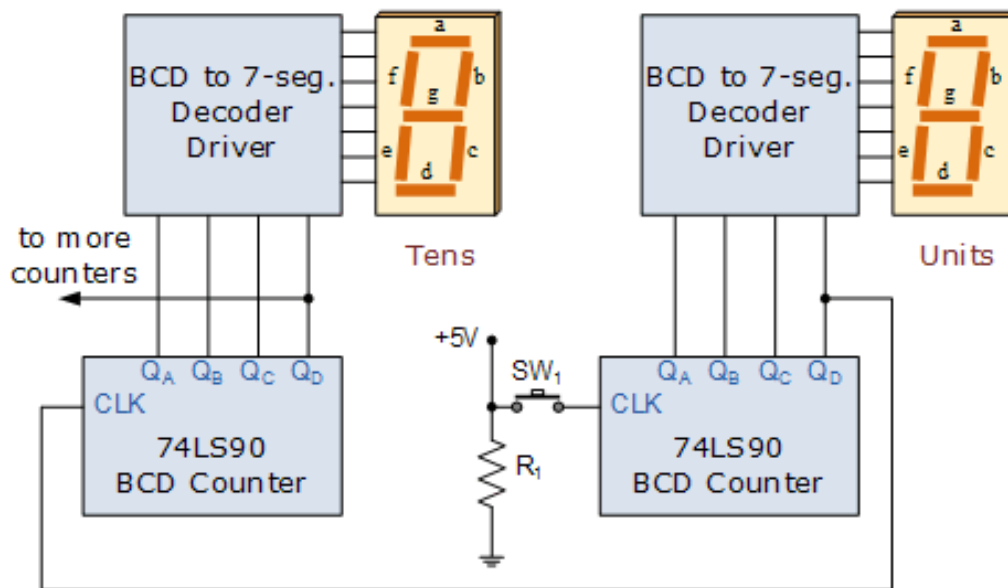


Figure A.3

Most 7 - segment displays are usually used in multi - digit counting application so by cascading together more BCD counter, 4 - digit counters giving displays with a maximum reading of 9999 can be constructed.

The 74LS90 BCD Counter is a very flexible counting circuit and can be used as a frequency divider or made to divide any whole number count from 2 to 9 by feeding the appropriate outputs back to the IC's Reset and Set inputs.

Appendix - B

8 - Bit 8 - Channel ADC0808

ADC0808 is a monolithic CMOS device, offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability and consumes minimal power.

ADC0808 is a converter that has eight analog inputs and eight digital outputs. ADC0808 allows us to monitor up to 8 different transducers through eight channels (IN0 to IN7) using only a single chip, the selection of the channel is decided by the address pins A, B, C. Fig. B.1 shows the pin diagram of ADC0808.

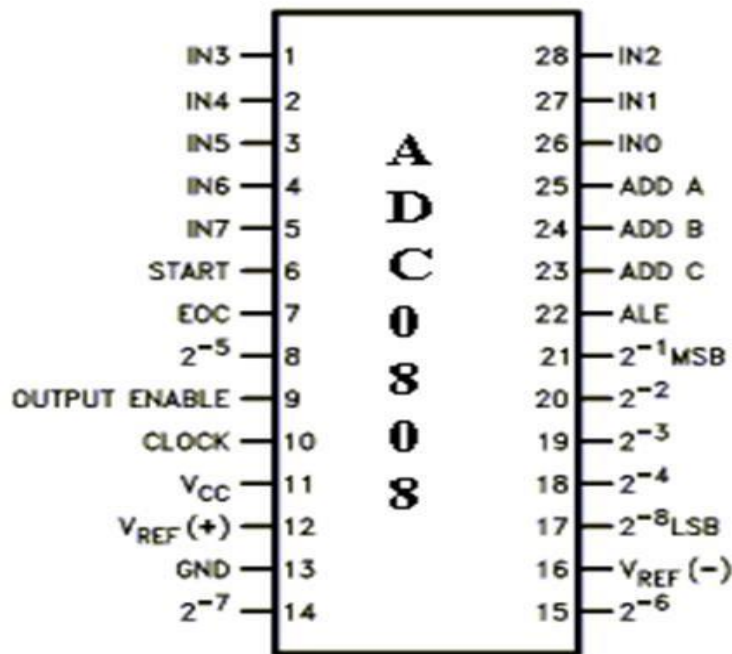


Figure B.1

Main Features:

- 0V to 5V input range.
- Easy interface to all μ Ps & μ Cs.
- Compatible with TTL voltage level specifications.
- Chip package with 28-pin.
- Resolution: 8 Bits
- Total unadjusted error: $\pm\frac{1}{2}$ LSB and ± 1 LSB
- Single supply: 5 VDC
- Low power consumption: 15 mW
- Conversion time: 100 μ s

The step size depends upon the set reference value, step size is the change in analog input to cause a unit change in the output of ADC. ADC0808 needs an external clock to operate, unlike single channel ADC0804 which has an internal clock.

When the start pin is held high no conversion begins, but when the start pin is made low the conversion will start within eight clock periods. At the point when the conversion is completed, the EOC pin goes low to indicate the completion of conversion process and data ready to be picked up. The output enables (OE) is then raised high, this enables the TRI-STATE outputs, allowing the data to be read.

Applications of Analog to Digital Converter:

- AC (air conditioner) includes temperature sensors to maintain the temperature within the room, this conversion of temperature can be achieved with the help of ADC.
- It is used in a digital oscilloscope to convert the signal from analog to digital to display.
- ADC is used to convert the analog voice signal to digital in mobile phones because mobile phones use digital voice signals but the voice signal is in the form of analog. So ADC is used to convert the signal before sending the signal toward the transmitter of the cell phone.
- ADC is used in medical devices like MRI and X-Ray to convert the images from analog to digital before alteration.
- The camera in the mobile is mainly used for capturing images as well as videos. These are stored in the digital device, so these are converted to digital form using ADC.

○

Appendix - C

Pin Diagrams of Important ICs

