Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

Laboratory Manual

(Computer Organization and Operating System MCA17403)

Table of Contents

Assignment/Lab Exercise/Activity No.	Name of the Experiment/Activity/Exercise
Assignment 1	Realization and implementation of the truth table of Basic gates (AND, OR, NOT)
Assignment 2	Realization and implementation of the truth table of Universal gates (NAND, NOR)
Assignment 3	Realization and implementation of the truth table of Exclusive gates (EX-OR, EX-NOR)
Assignment 4	Implementation of half adder
Assignment 5	Implementation of full adder
Assignment 6	Implementation of half subtractor
Assignment 7	Implementation of full subtractor
Assignment 8	Implementation of MUX
Assignment 9	Implementation of DEMUX
Assignment 10	Implementation of ENCODER
Assignment 11	Implementation of DECODER
Assignment 12	Implementation of Parallel Adder
Assignment 13	Implementation of Comparator
Assignment 14	Implementation of Arithmetic and Logic Unit (ALU)
Assignment 15	Implementation of Primary Memory (RAM)

Aim /Purpose of Assignments

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

The aim of an assignment involving the design of logic gates, adders, sub tractors, decoders, and encoders is to develop a fundamental understanding of digital logic design and its applications. These components are building blocks of digital circuits and are essential in the field of computer engineering and electronics.

Assignments 1-13

Assignment No: 1

(i) To Understand the basic principles of digital logic.

(ii) To Learn the functionality of different logic gates (AND, OR, NOT,

XOR, etc.). (iii)To design and implement simple logic expressions using logic

gates.

(iv) To explore the concept of truth tables and logic gate behavior.

Assignment No: 2

(i)To understand and design the basic gates using Universal gates NAND and NOR

Assignment No: 3

(i)To explore design and implement EX-OR and EX-NOR gates.

(ii) To design and implement EX-OR and Ex-NOR gates using Universal gates such NAND and NOR.

Assignment No: 4 and 5
(i)To Learn and design half-adders and full-adders
(ii) To explore the concept of truth tables of half adder and full –adders.
Assignment No: 6 and 7
(i)To Learn and design half-Sub tractor and full-Subtractor
(ii) To explore the concept of truth tables of half Subtractor and full –Subtractor..

Assignment No: 8

(i) To design a multiplexer (MUX) circuit that selects one of several input lines and routes it to the output.

(ii)To Grasp the concept of data selection and routing using multiplexers.

(iii)To Implement a multiplexer circuit using logic gates.

(iv)To Learn about applications of multiplexers in data routing, signal selection, and more.

Assignment No: 9

(i) To design a demultiplexer (DMUX) circuit that selects outputline for the corresponding input line.
(ii) To Grasp the concept of data selection and routing using demultiplexers.
(iii) To Implement a de- multiplexer circuit using logic gates.
(iv)To Learn about applications of de-multiplexers in data routing, signal selection and more.

Assignment No: 10
(i) To design an encoder circuit that encodes multiple input lines into a binary code.
(ii)To understand the process of encoding multiple inputs into a smaller set of outputs.
(iii)To implement an encoder using logic gates or other fundamental components.

Assignment No: 11
(i)To design a binary decoder circuit to decode a binary code into a set of output lines.
(ii)To grasp the concept of binary-to-decimal conversion.
(iii)To implement a binary decoder using logic gates or other basic components.

Assignment No: 12
(i)To design and implement a parallel adder circuit that can add two binary numbers together**.**
(ii)To understand the concept of binary addition and carry propagation.
(iii)To Implement a binary adder circuit using basic logic gates.

Assignment No: 13
(i) To design a binary comparator circuit that compares two binary numbers.
(ii)To understand the concept of binary number comparison.
(iii) To Implement a comparator circuit that outputs the relationship between two binary numbers.
(iv)To Learn about signed and unsigned number comparison.

Assignment No: 14
(i)The aim of this verification process is to understand, demonstrate, and validate the fundamental operating procedures of ALU chip.

Assignment No: 15
(i)The aim of this verification process is to understand, demonstrate, and validate the fundamental operating procedures involved in reading from and writing to RAM ICs, as well as to comprehend the technique of cascading multiple RAM ICs to expand the memory space.

Learning Outcomes

(i) Analyze the behaviour of logic gates.
(ii) The students will understand various types of basic & universal logic gates.
(iii) The students will learn how to design Combinational circuits.
(iv) To design Arithmetic logic units and different types of memory blocks.
(v) To develop logic for assembly language programming.

Prerequisites: Knowledge of logic circuits -combinational and sequential.

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

Software required:
Online Simulator

Introduction and Theory

Assignment 1

Logic gates are idealized or physical devices implementing a Boolean function, which it performs a logical operation on one or more logical inputs and produce a single output. Depending on the context, the term may refer to an ideal logic gate, one that has for instance zero rise time and unlimited fan out or it may refer to a non-ideal physical device.

The main hierarchy is as follows: -

- Basic Gates
- Universal Gates
- Advanced Gates

1.**AND gate: -** Function of AND gate is to give the output true when both the inputs are true. In all the other remaining cases output becomes false. Following table justifies the statement: -

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

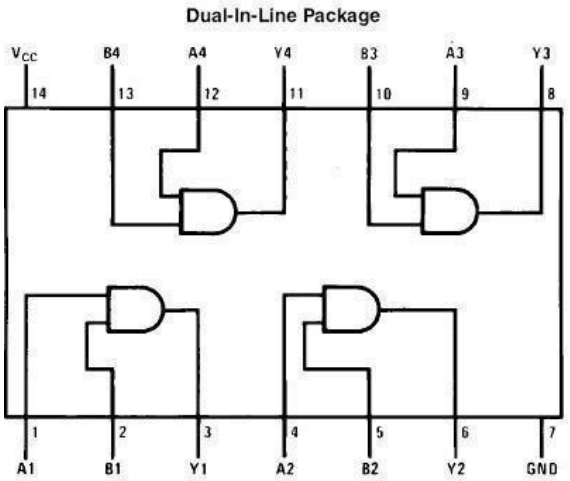


Fig 1 of AND gate IC 7408

2.**OR gate: -** Function of OR gate is to give output true when one of the either inputs are true. In the remaining case output becomes false. Following table justify the statement: -

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

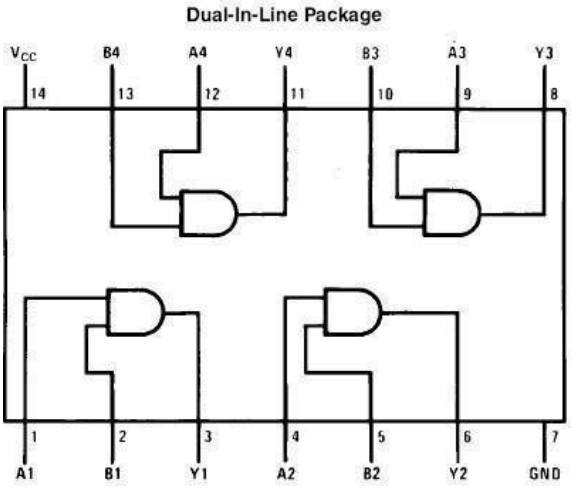


Fig2:OR GateIC 7432

3.**NOT gate: -** Function of NOR gate is to reverse the nature of the input. It converts true input to false and vice versa. Following table justifies the statement:-

Input	Output
1	0
0	1

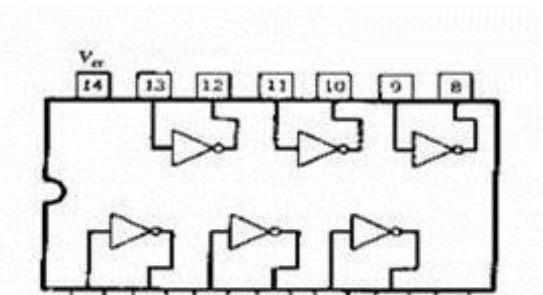


Fig 3 NOT GateIC7404

Assignment No:2

Universal Gate

NAND gate: - Function of NAND gate is to give true output when one of the two provided input are false. In the remaining output is true case. Following table justifies the statement :-

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

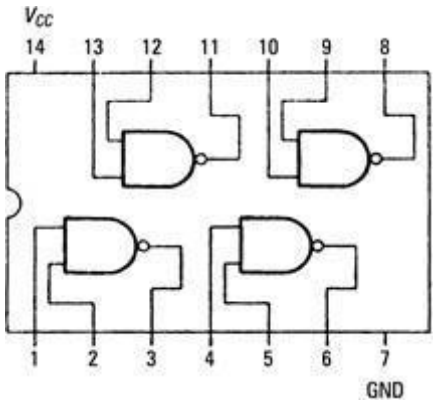


Fig 4: NAND Gate IC 7400

NOR gate: - NOR gate gives the output true when both the two provided input are false. In all the other cases output remains false. Following table justifies the statement:-

Input A	Input B	Output
1	1	0

1	0	0
0	1	0
0	0	1

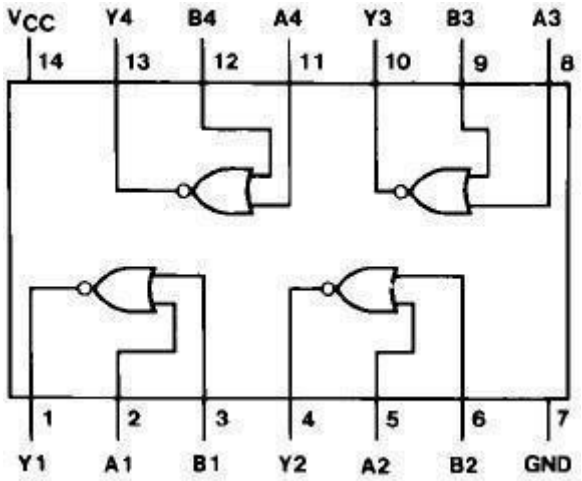


Fig:5 NOR GateIC 7402

Assignment 3

XOR gate: - The function of XOR gate is to give output true only when both the inputs are true. Following table explains this:-

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

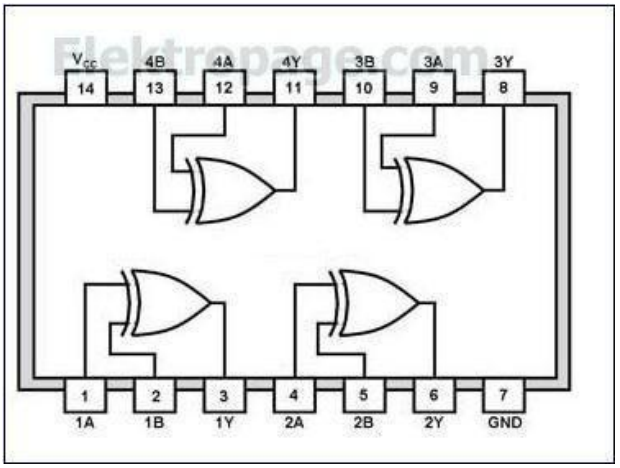


Fig :6 XOR GateIC74136

XNOR gate: - The function of XOR gate is to give output true only when both the inputs are true. Following table explains this:-

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

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

Assignment 4

An **Adder** is a device that can add two binary digits. It is a type of digital circuit that performs the operation of additions of two number. It is mainly designed for the addition of binary number, but they can be used in various other applications like binary code decimal, address decoding, table index calculation, etc.
There are two types of Adder:

- Half Adder
- Full Adder

Half Adder:

There are two inputs and two outputs in a Half Adder. Inputs are named as A and B, and the outputs are named as Sum (S) and Carry (C). The Sum is X-OR of the input A and B. Carry is AND of the input A and B. With the help of half adder, one can design a circuit that is capable of performing simple addition with the help of logic gates.

The **truth table** of the half adder is shown below:

Input	Input	Output	
A	**B**	**sum**	**carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

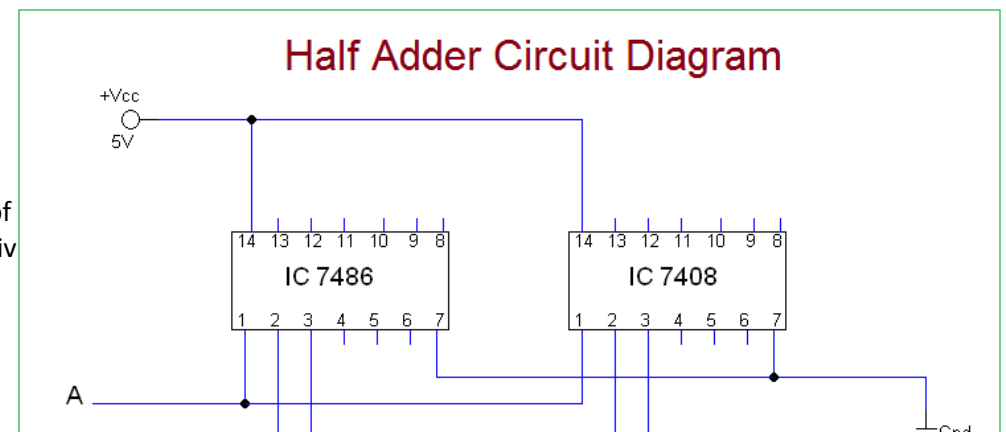


Fig 7: Half Adder Circuit

Assignment 5

Full Adder:

The full adder is a little more difficult to implement than a half adder. The main difference between a half adder and a full adder is that the full-adder has three inputs and two outputs. The two inputs are A and B, and the third input is a carry input C_{IN}. The output carry is designated as C_{OUT}, and the normal output is designated as S.

The **truth table** of the Full Adder Circuit is shown below:

Input	Output

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

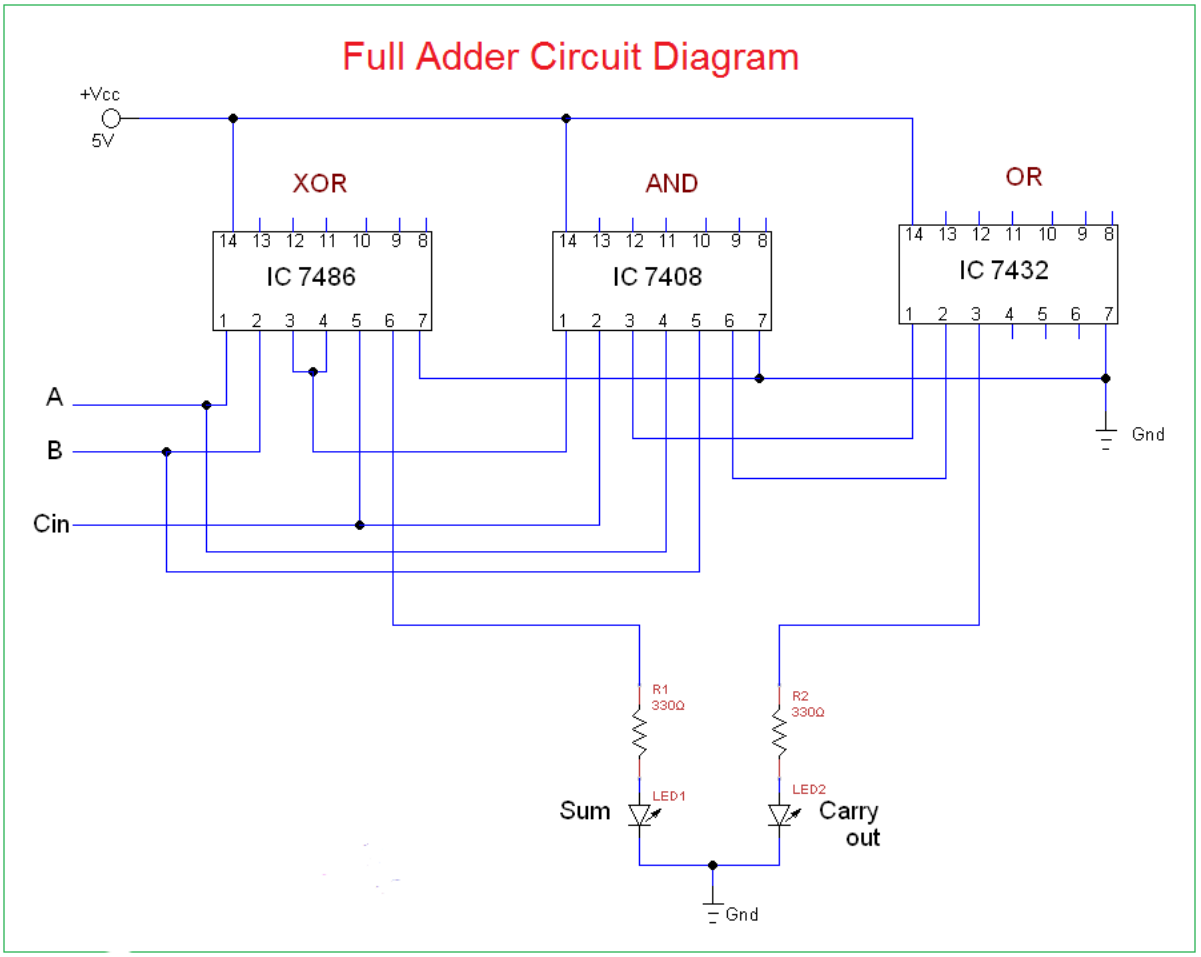


Fig :8 Full Adder Circuit

-

Assignment 6

Half Subtractor:

A half subtractor is a combinational circuit that subtracts two single-bit binary numbers and produces their difference along with a borrow-out. Here's the truth table*:*

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

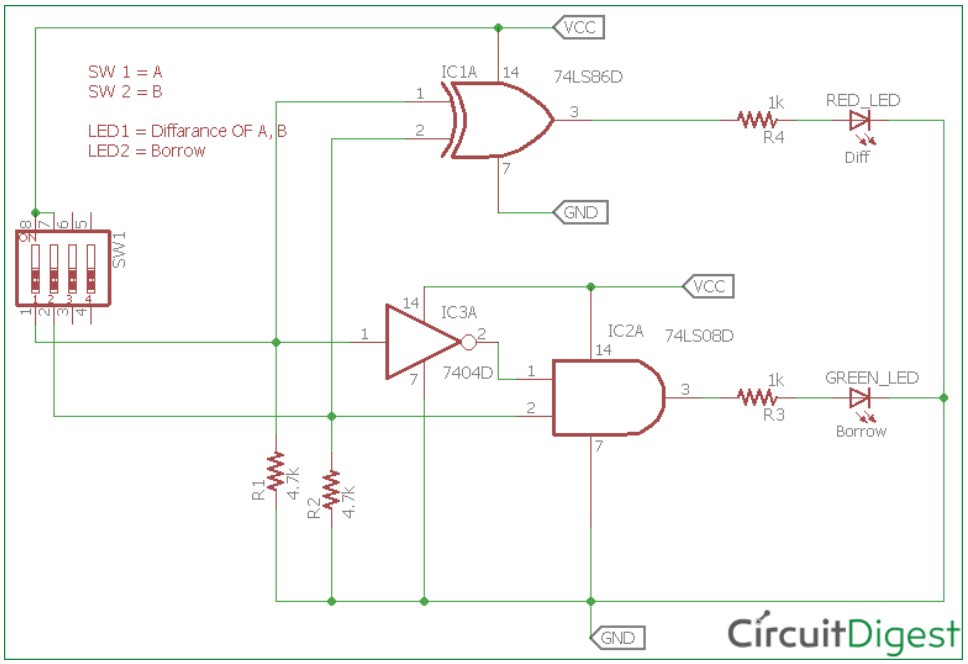


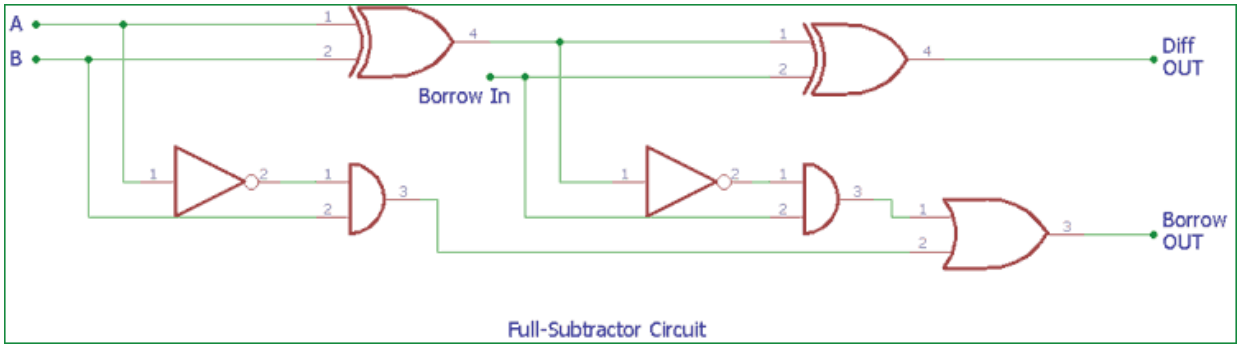
Fig:9 Half Subtractor

Assignment 7

Full Subtractor:

A full subtractor is a combinational circuit that subtracts three single-bit binary numbers (minuend, subtrahend, and borrow-in) and produces their difference along with a borrow-out. Here's the truth table:

A	B	Borrow-In	Difference	Borrow-Out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1



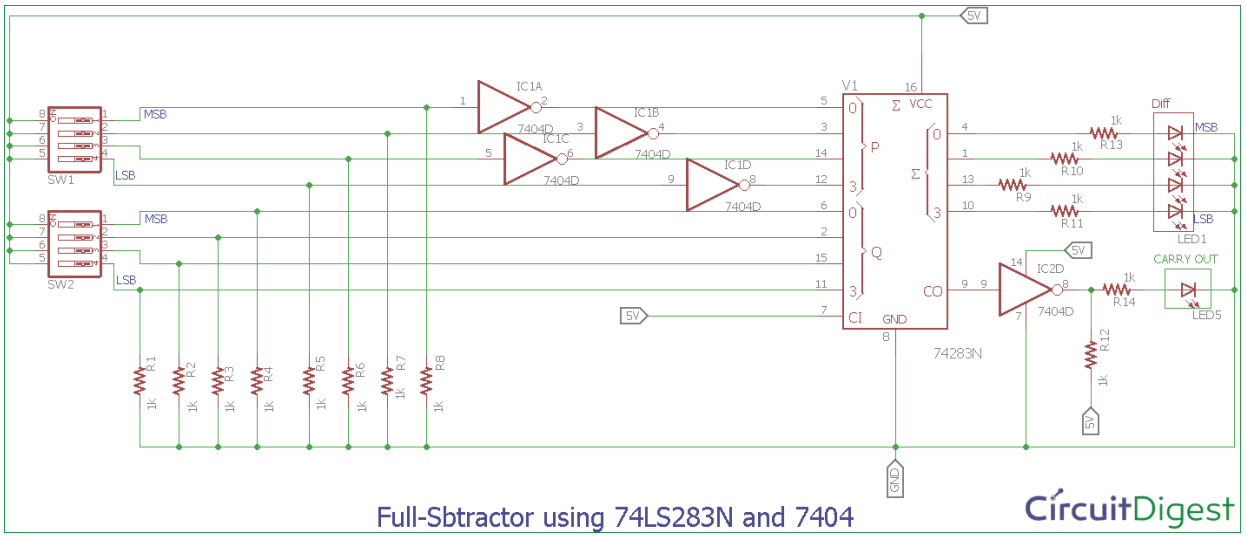


Fig 10: Circuit Diagram of Full Subtractor

Assignment 8

A 4:1 dual multiplexer refers to a digital circuit that takes in four input signals and two selection signals, and produces a single output signal based on the combination of these inputs.

The "4:1" aspect indicates that there are four data inputs, labeled D0, D1, D2, and D3. These inputs carry the data that you want to select from.

The "dual" aspect typically means that there are two separate instances of this 4:1 multiplexer circuit within the same chip or module. Each of these instances has its own set of data inputs (D0 to D3) and selection signals.

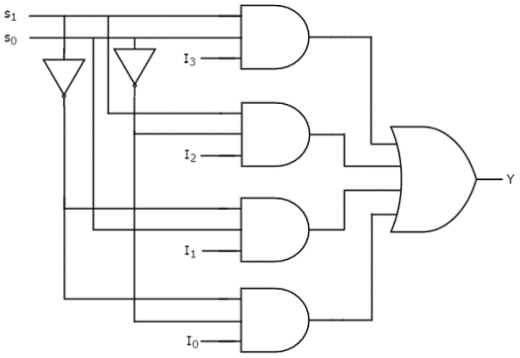
A multiplexer works based on the values of the selection signals. In a 4:1 multiplexer, there are two selection signals, typically labeled S0 and S1. These two signals determine which data input is passed to the output.

The logic for a 4:1 multiplexer can be described as follows:

If S1 and S0 are both 0, the output is D0.

If S1 is 0 and S0 is 1, the output is D1.
If S1 is 1 and S0 is 0, the output is D2.
If S1 and S0 are both 1, the output is D3.
By having two instances of this 4:1 multiplexer circuit, you effectively have the capability to select between eight different input signals using the two sets of selection signals.

Multiplexers are commonly used in digital circuits for various purposes, including data routing, signal selection, and controlling multiple devices using fewer control lines. They play a crucial role in digital system design and implementation.



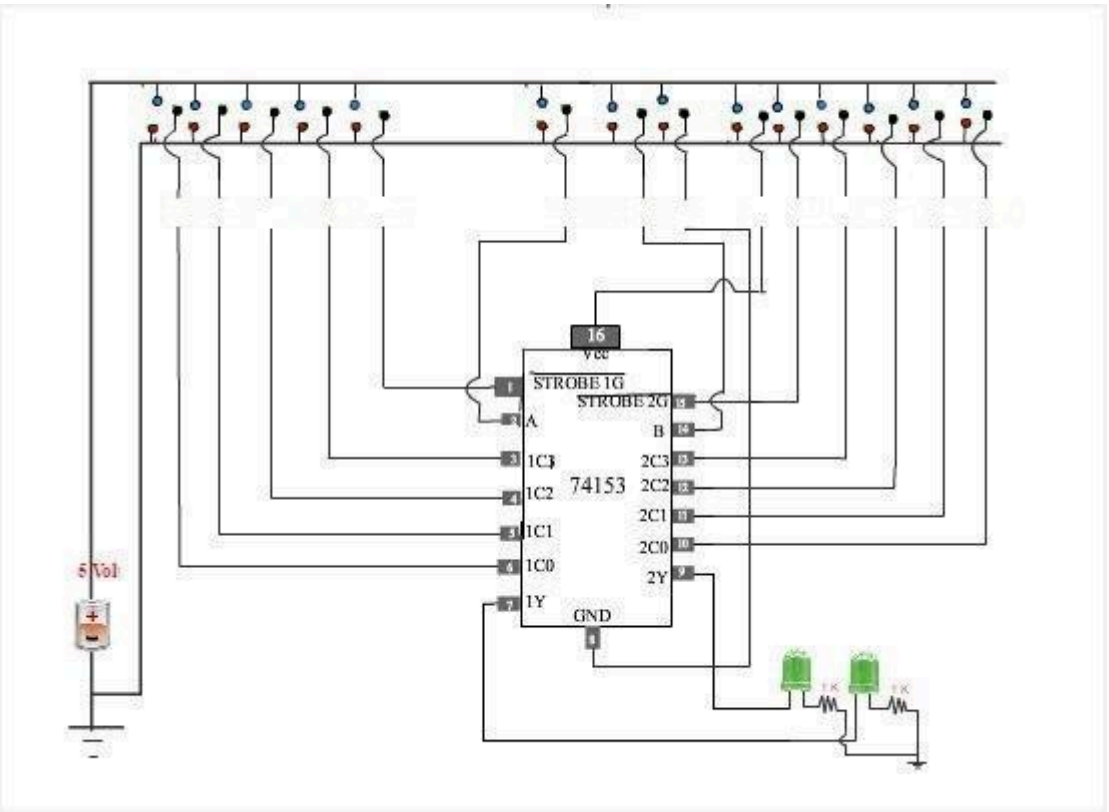


Fig11: Circuit Diagram of 4:1 Multiplexer

Assignment 9

The process of getting information from one input and transmitting the same over one of many outputs is called Demultiplexing. If you recall the Multiplexer tutorial, there we discussed the concept of Multiplexing. Demultiplexing is just the opposite of that. A Demultiplexer is a combinational logic circuit that receives the information on a single input line and transmits the same information over one of 'n' possible output lines.

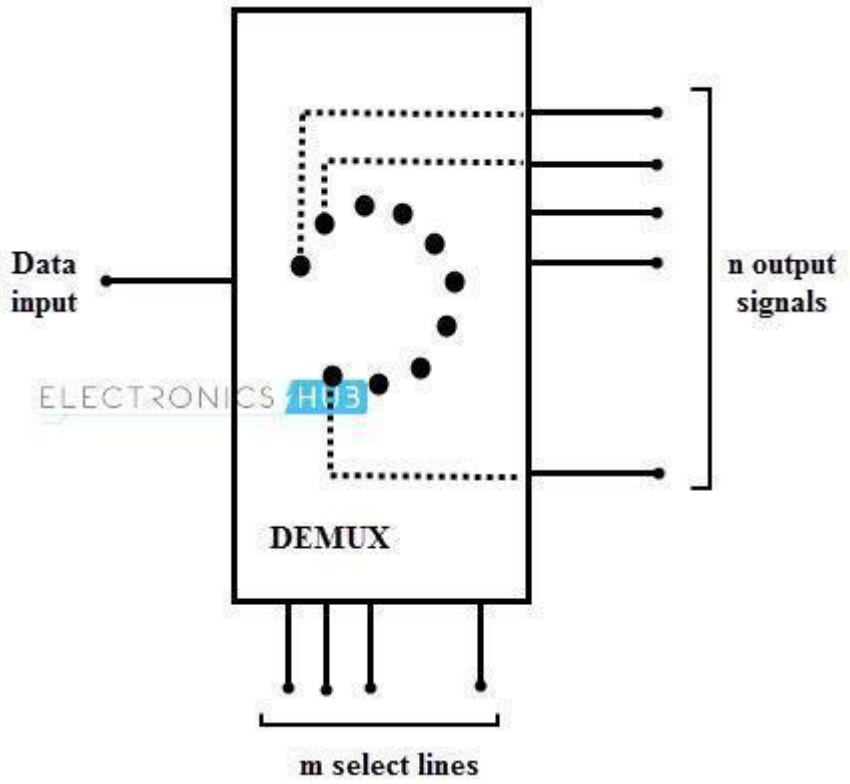
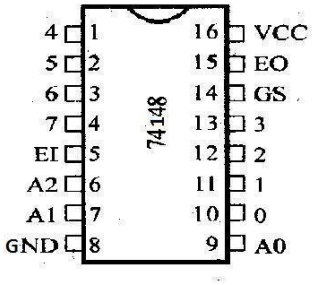


Fig :1 Block Diagram Demultiplexer

These are available in different IC packages and some of the most commonly used demultiplexer ICs includes 74139 (dual 1:4 DEMUX), 74138 (1:8 DEMUX), 74237 (1:8 DEMUX with Address Latches), 74154 (1:16 DEMUX), 74159 (1:16 DEMUX open collector type), etc.
The Demultiplexer ICs are also called as Decoder ICs. For example, 74159 is a 4-line to 16-line Decoder IC.

Assignment 10
Encoder

IC 74148 8 Line to 3 Line Encoder



INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	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	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

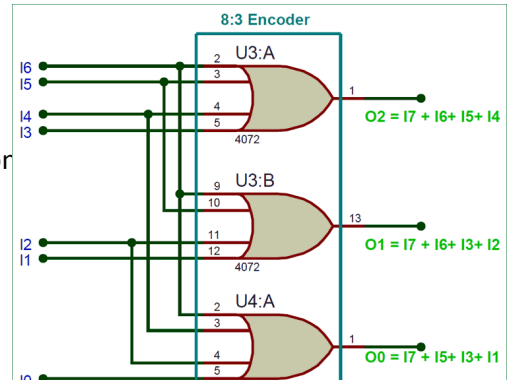


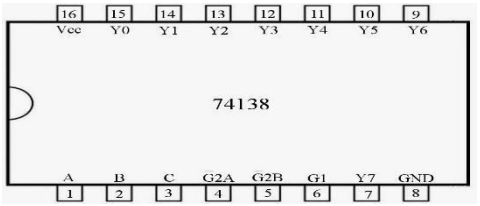
Fig 12 Circuit diagram of Encoder

Pin No.	Pin Name	Details
1	4	Input
2	5	Input
3	6	Input
4	7	Input (MSB)
5	EI	Enable Input
6	A2	Output
7	A1	Output
8	GND	Ground
9	A0	Output
10	0	Input (LSB)
11	1	Input
12	2	Input
13	3	Input

14	GS	Select Output
15	EO	Enable Output
16	Vcc	Power supply

Assignment 11

Decoder



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

H - HIGH, L - LOW, X - DONT CARE

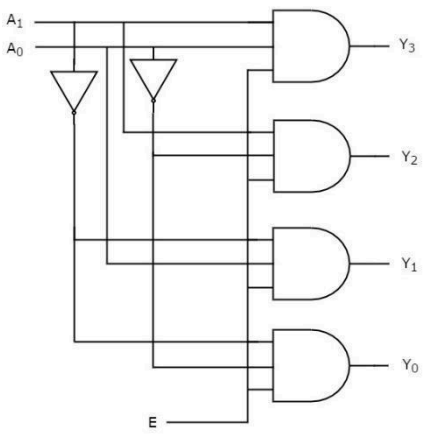


Fig 13: Circuit Diagram of Decoder

Assignment 12
Parallel Adder

IC 7483 4 Bit Parallel Adder

The 7483 is the most popular 4-bit carry look ahead (CLA) adder:

```
    A4  A3  A2  A1
    B4  B3  B2  B1
 ----------------------------
C4  S4  S3  S2  S1  C1
```

Here, A4, A3, A2, A1 and B4, B3, B2, B1 are the two input data and S4, S3, S2, S1 is the sum and C4 is the resultant carry. Where, C1 is the carry input to the least significant position. The two carry input/output enable this chip to be cascaded.

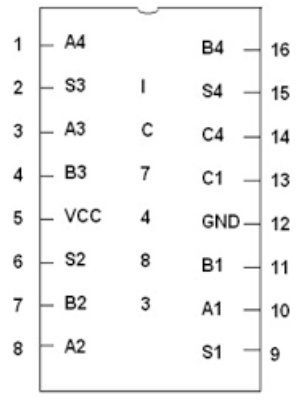


Fig:14 IC7483 4-bit Parallel Adder

Description of all the pins and its functions:

Pin No.	Pin Name	Pin Details
1	A4	MSB of Input A(4the bit)
2	S3	3rd LSB of sum output S (3rd bit)
3	A3	3rd LSB of Input A (3rd bit)
4	B3	3rd LSB of Input V (3rd bit)
5	Vcc	Power supply
6	S2	2dd LSB of sum output S (2nd bit)
7	B2	2nd LSB of Input B (2nd bit)
8	A2	2nd LSB of Input A (2nd bit)
9	S1	LSB of sum output s (1st bit)
10	B1	LSB of Input B (1st bit)
11	A1	LSB of Input A (1st bit)
12	GND	Ground

13	C1	Carry input
14	C4	Resultant carry output
15	S4	MSB of sum output S (4th bit)
16	B4	MSB of Input B (4th bit)

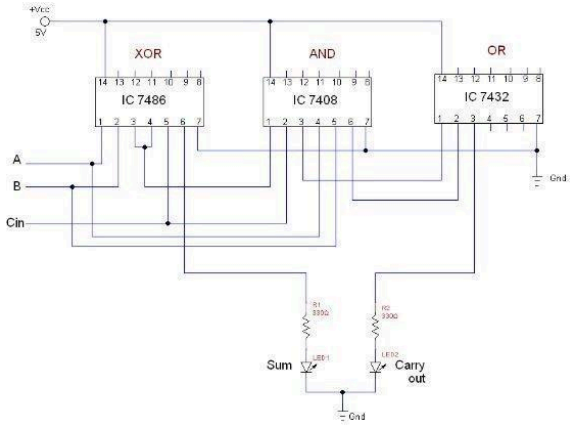


Fig:15 4 bit Parallel Adder

Assignment 13

Comparator

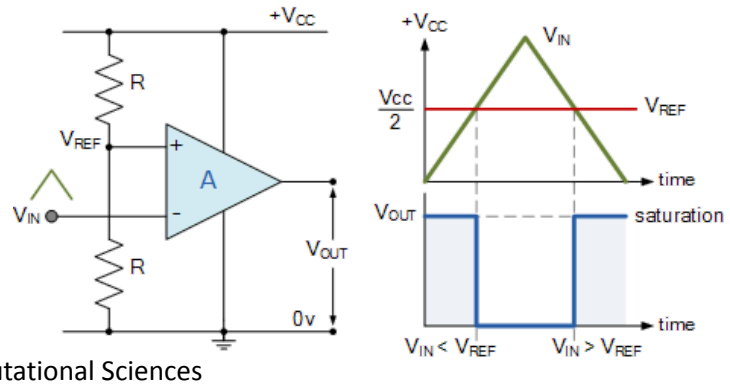


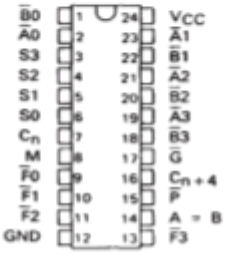
Fig:16 Comparator Circuit

A comparator is an electronic circuit or device used to compare two analog voltage signals and determine their relationship in terms of magnitude. It essentially performs a binary decision, indicating whether one input voltage is greater than the other. The output of a comparator is typically a digital signal that represents this comparison result

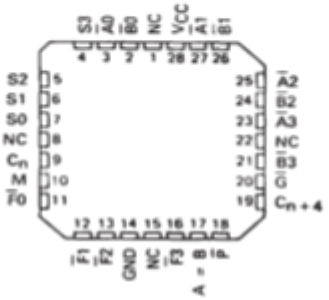
Assignment 14

- Full Look-Ahead for High-Speed Operations on Long Words
- Input Clamping Diodes Minimize Transmission-Line Effects
- Darlington Outputs Reduce Turn-Off Time
- Arithmetic Operating Modes:
  - Addition
  - Subtraction
  - Shift Operand A One Position
  - Magnitude Comparison
  - Plus Twelve Other Arithmetic Operations
- Logic Function Modes:
  - Exclusive-OR
  - Comparator
  - AND, NAND, OR, NOR
  - Plus Ten Other Logic Operations

SN54LS181, SN54S181 . . . J OR W PACKAGE
SN74LS181, SN74S181 . . . DW OR N PACKAGE
(TOP VIEW)



SN54LS181, SN54S181 . . . FK PACKAGE
(TOP VIEW)



NC - No internal connection

TYPICAL ADDITION TIMES

NUMBER OF BITS	ADDITION TIMES		PACKAGE COUNT		CARRY METHOD BETWEEN ALUs
	USING 'LS181 AND 'S182	USING 'S181 AND 'S182	ARITHMETIC/ LOGIC UNITS	LOOK-AHEAD CARRY GENERATORS	
1 to 4	24 ns	11 ns	1		NONE
5 to 8	40 ns	18 ns	2		RIPPLE
9 to 16	44 ns	19 ns	3 or 4	1	FULL LOOK-AHEAD
17 to 64	68 ns	28 ns	5 to 16	2 to 5	FULL LOOK-AHEAD

description

The 'LS181 and 'S181 are arithmetic logic units (ALU)/function generators that have a complexity of 75 equivalent gates on a monolithic chip. These circuits perform 16 binary arithmetic operations on two 4-bit words as shown in Tables 1 and 2. These operations are selected by the four function-select lines (S0, S1, S2, S3) and include addition, subtraction, decrement, and straight transfer. When performing arithmetic manipulations, the internal carries must be enabled by applying a low-level voltage to the mode control input (M). A full carry look-ahead scheme is made available in these devices for fast, simultaneous carry generation by means of two cascade-outputs (pins 15 and 17) for the four bits in the package. When used in conjunction with the SN54S182 or SN74S182 full carry look-ahead circuits, high-speed arithmetic operations can be performed. The typical addition times shown above illustrate the little additional time required for addition of longer words when full carry look-ahead is employed. The method of cascading 'S182 circuits with these ALUs to provide multi-level full carry look-ahead is illustrated under typical applications data for the 'S182.

If high speed is not of importance, a ripple-carry input (C_n) and a ripple-carry output (C_{n+4}) are available. However, the ripple-carry delay has also been minimized so that arithmetic manipulations for small word lengths can be performed without external circuitry.

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

Department of Computational Sciences
Brainware University

description (continued)

The 'LS181 and 'S181 will accommodate active-high data if the pin designations are interpreted as follows:

PIN NUMBER	2	1	23	22	21	20	19	18	9	10	11	13	7	16	15	17
Active-low data (Table 1)	$\overline{A}_0$	$\overline{B}_0$	$\overline{A}_1$	$\overline{B}_1$	$\overline{A}_2$	$\overline{B}_2$	$\overline{A}_3$	$\overline{B}_3$	$\overline{F}_0$	$\overline{F}_1$	$\overline{F}_2$	$\overline{F}_3$	C_n	C_{n+4}	$\overline{P}$	$\overline{G}$
Active-high data (Table 2)	A_0	B_0	A_1	B_1	A_2	B_2	A_3	B_3	F_0	F_1	F_2	F_3	$\overline{C}_n$	$\overline{C}_{n+4}$	X	Y

Subtraction is accomplished by 1's complement addition where the 1's complement of the subtrahend is generated internally. The resultant output is A—B—1, which requires an end-around or forced carry to provide A—B.

The 'LS181 or 'S181 can also be utilized as a comparator. The A = B output is internally decoded from the function outputs (F0, F1, F2, F3) so that when two words of equal magnitude are applied at the A and B inputs, it will assume a high level to indicate equality (A = B). The ALU must be in the subtract mode with C_n = H when performing this comparison. The A = B output is open-collector so that it can be wire-AND connected to give a comparison for more than four bits. The carry output (C_{n+4}) can also be used to supply relative magnitude information. Again, the ALU must be placed in the subtract mode by placing the function select inputs S3, S2, S1, S0 at L, H, H, L, respectively.

INPUT C_n	OUTPUT C_{n+4}	ACTIVE-LOW DATA (FIGURE 1)	ACTIVE-HIGH DATA (FIGURE 2)
H	H	A ≥ B	A ≤ B
H	L	A < B	A > B
L	H	A > B	A < B
L	L	A ≤ B	A ≥ B

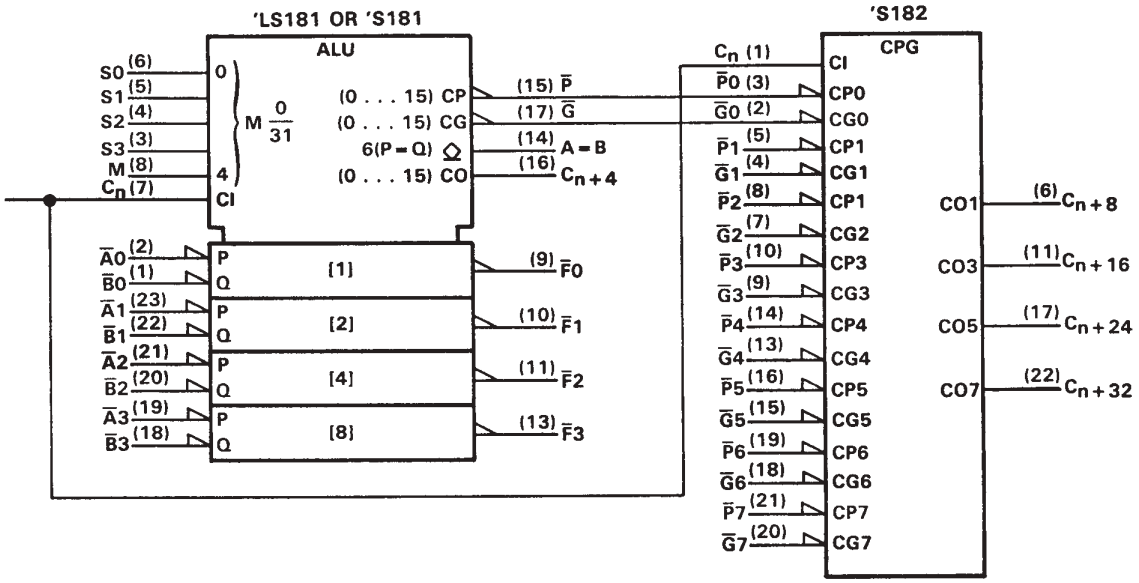
These circuits have been designed to not only incorporate all of the designer's requirements for arithmetic operations, but also to provide 16 possible functions of two Boolean variables without the use of external circuitry. These logic functions are selected by use of the four function-select inputs (S0, S1, S2, S3) with the mode-control input (M) at a high level to disable the internal carry. The 16 logic functions are detailed in Tables 1 and 2 and include exclusive-OR, NAND, AND, NOR, and OR functions.

Series 54, 54LS, and 54S devices are characterized for operation over the full military temperature range of −55°C to 125°C; Series 74LS and 74S devices are characterized for operation from 0°C to 70°C.

signal designations

In both Figures 1 and 2, the polarity indicators (◺) indicate that the associated input or output is active-low with respect to the function shown inside the symbol, and the symbols are the same in both figures. The signal designations in Figure 1 agree with the indicated internal functions based on active-low data, and are for use with the logic functions and arithmetic operations shown in Table 1. The signal designations have been changed in Figure 2 to accommodate the logic functions and arithmetic operations for the active-high data given in Table 2. The 'LS181 and 'S181, together with the 'S182, can be used with the signal designation of either Figure 1 or Figure 2.

logic symbols† and signal designations (active-low data)



†These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.
Pin numbers shown are for dual-in-line and "small outline" packages.

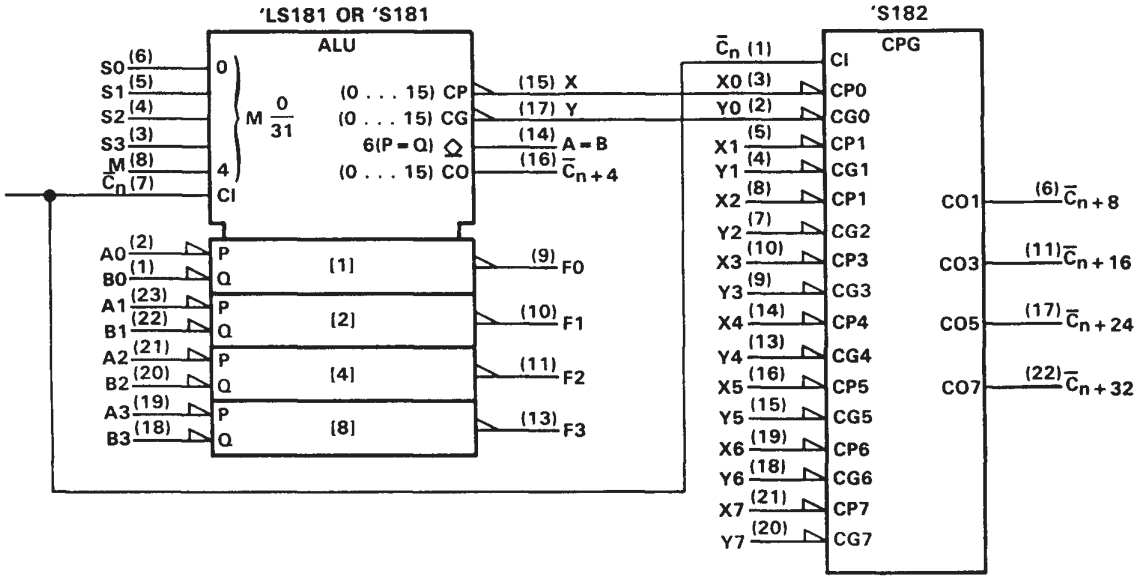
FIGURE 1 (USE WITH TABLE 1)

TABLE 1

SELECTION				ACTIVE-LOW DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	Cn = L (no carry)	Cn = H (with carry)
L	L	L	L	F = $\overline{A}$	F = A MINUS 1	F = A
L	L	L	H	F = $\overline{AB}$	F = AB MINUS 1	F = AB
L	L	H	L	F = $\overline{A}$ + B	F = A$\overline{B}$ MINUS 1	F = A$\overline{B}$
L	L	H	H	F = 1	F = MINUS 1 (2's COMP)	F = ZERO
L	H	L	L	F = $\overline{A + B}$	F = A PLUS (A + $\overline{B}$)	F = A PLUS (A + $\overline{B}$) PLUS 1
L	H	L	H	F = $\overline{B}$	F = AB PLUS (A + $\overline{B}$)	F = AB PLUS (A + $\overline{B}$) PLUS 1
L	H	H	L	F = $\overline{A \oplus B}$	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	F = A + $\overline{B}$	F = A + $\overline{B}$	F = (A + $\overline{B}$) PLUS 1
H	L	L	L	F = $\overline{A}$B	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
H	L	L	H	F = A $\oplus$ B	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = A$\overline{B}$ PLUS (A + B)	F = A$\overline{B}$ PLUS (A + B) PLUS 1
H	L	H	H	F = A + B	F = (A + B)	F = (A + B) PLUS 1
H	H	L	L	F = 0	F = A PLUS A‡	F = A PLUS A PLUS 1
H	H	L	H	F = A$\overline{B}$	F = AB PLUS A	F = AB PLUS A PLUS 1
H	H	H	L	F = AB	F = A$\overline{B}$ PLUS A	F = A$\overline{B}$ PLUS A PLUS 1
H	H	H	H	F = A	F = A	F = A PLUS 1

‡Each bit is shifted to the next more significant position.

logic symbols† and signal designations (active-high data)



†These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.
Pin numbers shown are for dual-in-line and "small outline" packages.

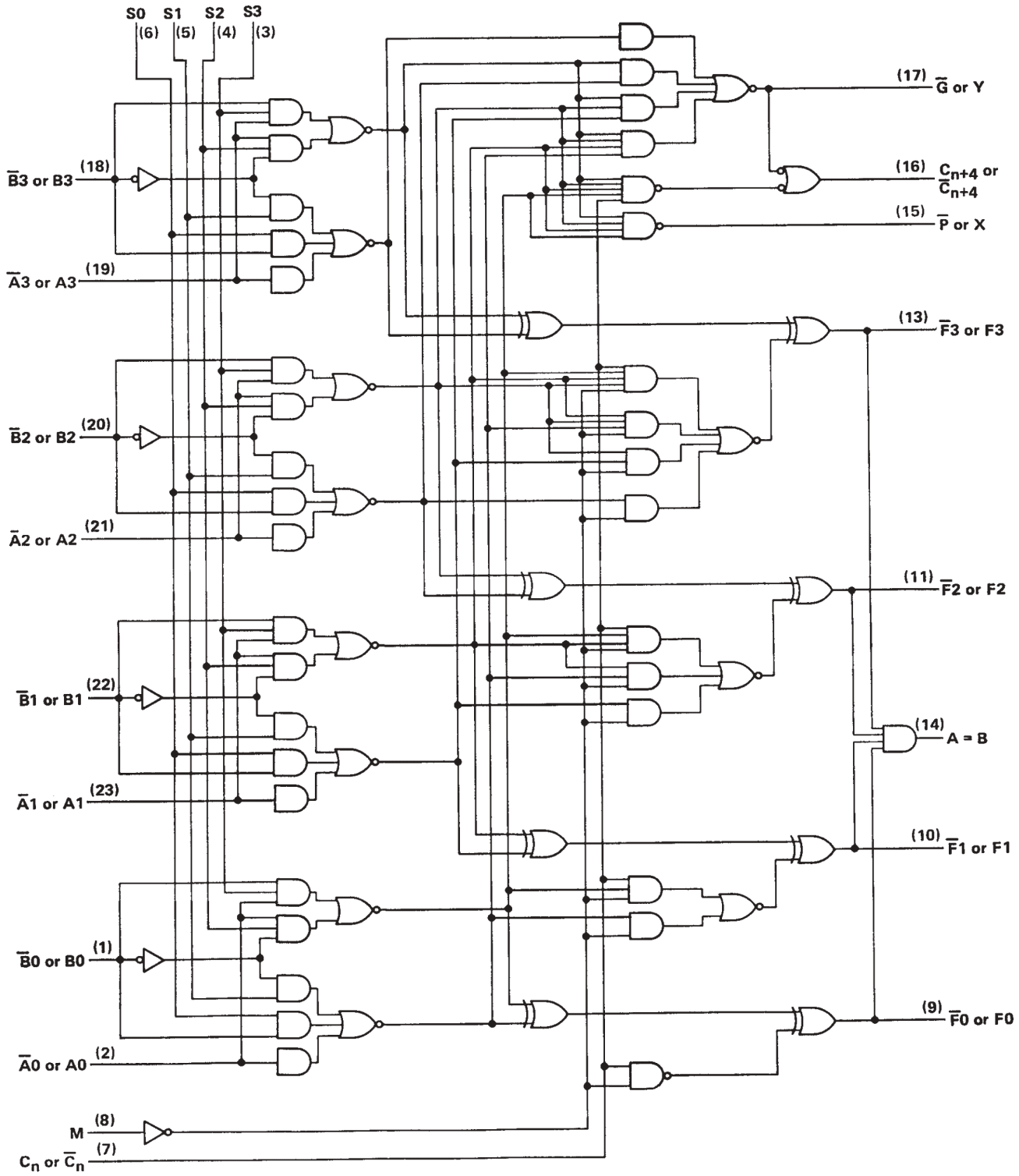
FIGURE 2 (USE WITH TABLE 2)

TABLE 2

SELECTION				ACTIVE-HIGH DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	$\overline{C_n}$ = H (no carry)	$\overline{C_n}$ = L (with carry)
L	L	L	L	F = $\overline{A}$	F = A	F = A PLUS 1
L	L	L	H	F = $\overline{A+B}$	F = A + B	F = (A + B) PLUS 1
L	L	H	L	F = $\overline{A}B$	F = A + $\overline{B}$	F = (A + $\overline{B}$) PLUS 1
L	L	H	H	F = 0	F = MINUS 1 (2's COMPL)	F = ZERO
L	H	L	L	F = $\overline{AB}$	F = A PLUS A$\overline{B}$	F = A PLUS A$\overline{B}$ PLUS 1
L	H	L	H	F = $\overline{B}$	F = (A + B) PLUS A$\overline{B}$	F = (A + B) PLUS A$\overline{B}$ PLUS 1
L	H	H	L	F = A ⊕ B	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	F = A$\overline{B}$	F = A$\overline{B}$ MINUS 1	F = A$\overline{B}$
H	L	L	L	F = $\overline{A}$ + B	F = A PLUS AB	F = A PLUS AB PLUS 1
H	L	L	H	F = $\overline{A \oplus B}$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = (A + $\overline{B}$) PLUS AB	F = (A + $\overline{B}$) PLUS AB PLUS 1
H	L	H	H	F = AB	F = AB MINUS 1	F = AB
H	H	L	L	F = 1	F = A PLUS A†	F = A PLUS A PLUS 1
H	H	L	H	F = A + $\overline{B}$	F = (A + B) PLUS A	F = (A + B) PLUS A PLUS 1
H	H	H	L	F = A + B	F = (A + $\overline{B}$) PLUS A	F = (A + $\overline{B}$) PLUS A PLUS 1
H	H	H	H	F = A	F = A MINUS 1	F = A

† Each bit is shifted to the next more significant position.

logic diagram (positive logic)



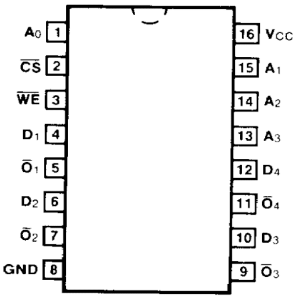
Pin numbers shown are for DW, J, N, and W packages.

Assignment 15

54S/74S189 011745
54LS/74LS189 611750

64-BIT RANDOM ACCESS MEMORY
(With 3-State Outputs)

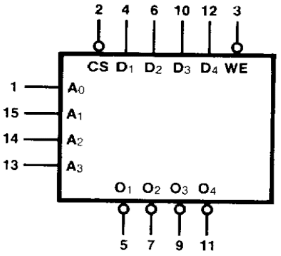
CONNECTION DIAGRAM
PINOUT A



DESCRIPTION — The '189 is a high speed 64-bit RAM organized as a 16-word by 4-bit array. Address inputs are buffered to minimize loading and are fully decoded on-chip. The outputs are 3-state and are in the high impedance state whenever the Chip Select ($\overline{CS}$) input is HIGH. The outputs are active only in the Read mode and the output data is the complement of the stored data.

- **3-STATE OUTPUTS FOR DATA BUS APPLICATIONS**
- **BUFFERED INPUTS MINIMIZE LOADING**
- **ADDRESS DECODING ON-CHIP**
- **DIODE CLAMPED INPUTS MINIMIZE RINGING**

LOGIC SYMBOL



VCC = Pin 16
GND = Pin 8

ORDERING CODE: See Section 9

PKGS	PIN OUT	COMMERCIAL GRADE VCC = +5.0 V ±5%, TA = 0°C to +70°C	MILITARY GRADE VCC = +5.0 V ±10%, TA = -55°C to +125°C	PKG TYPE
Plastic DIP (P)	A	74S189PC, 74LS189PC		9B
Ceramic DIP (D)	A	74S189DC, 74LS189DC	54S189DM, 54LS189DM	6B
Flatpak (F)	A	74S189FC, 74LS189FC	54S189FM, 54LS189FM	4L

INPUT LOADING/FAN-OUT: See Section 3 for U.L. definitions

PIN NAMES	DESCRIPTION	54/74S (U.L.) HIGH/LOW	54/74LS (U.L.) HIGH/LOW
$A_0 - A_3$	Address Inputs	0.63/0.16	0.5/0.013
$\overline{CS}$	Chip Select Input (Active LOW)	0.63/0.16	0.5/0.013
$\overline{WE}$	Write Enable Input (Active LOW)	0.63/0.16	0.5/0.013
$D_1 - D_4$	Data Inputs	0.63/0.16	0.5/0.013
$\overline{O}_1 - \overline{O}_4$	Inverted Data Outputs	162/10 (50)	10/10 (5.0)

4

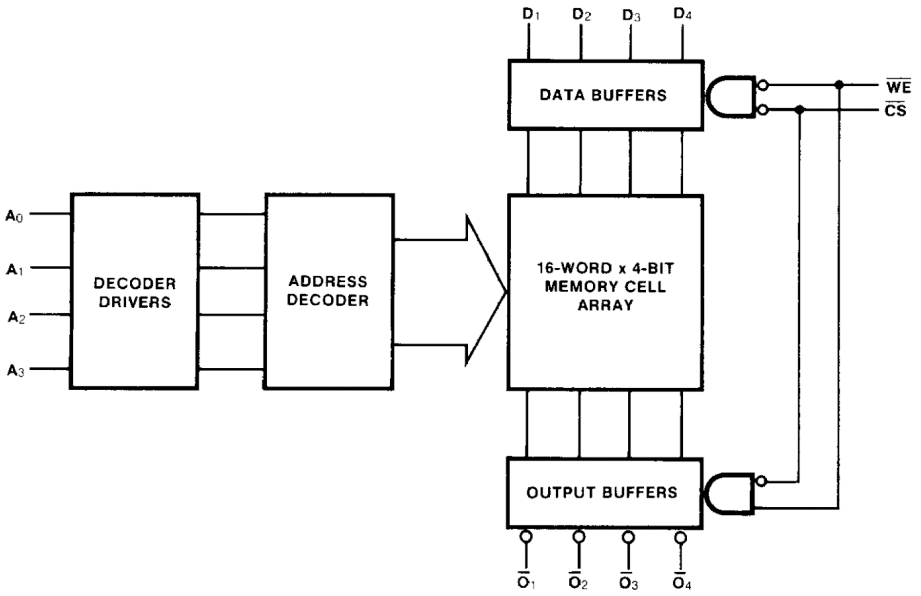
189

FUNCTION TABLE

INPUTS $\overline{CS}$	INPUTS $\overline{WE}$	OPERATION	CONDITION OF OUTPUTS
L	L	Write	High Impedance
L	H	Read	Complement of Stored Data
H	X	Inhibit	High Impedance

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

LOGIC DIAGRAM



DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		54/74S Min	54/74S Max	54/74LS Min	54/74LS Max	UNITS	CONDITIONS
V_{OL}	Output LOW Voltage	XM		0.5		0.4	V	V_{CC} = Min I_{OL} = 16 mA ('S189) I_{OL} = 8.0 mA (54LS189) I_{OL} = 16 mA (74LS189)
		XC		0.45		0.5		
V_{OH}	Output HIGH Voltage	XM		2.4		2.8	V	V_{CC} = Min I_{OH} = 2.0 mA (54S189) I_{OH} = 6.5 mA (74S189) I_{OH} = 0.4 mA ('LS189)
		XC		2.4		2.8		
I_{OS}	Output Short Circuit Current		-30	-100	-80*		mA	V_{CC} = Max
I_{CC}	Power Supply Current			110		40	mA	V_{CC} = Max; $\overline{WE}$, $\overline{CS}$, Gnd

*Typical Value

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

Assignment 1:

(i)What is the purpose of an AND gate?
(ii)Give an example of a real-world situation that can be represented using an AND gate.
(iii)Explain the behavior of an OR gate.
(iv)Give an example of a real-world situation that can be represented using an OR gate.
(v)What does a NOT gate do?
(vi)Give an example of a real-world situation that can be represented using a NOT gate.

Assignment2

(i) How do NAND and NOR gates compare to other basic logic gates like AND and OR gates in terms of versatility a
(ii)Can you provide an example of implementing a non-trivial logic function using only NAND gates?
(iii)How can you create a NOT gate using a single NAND gate?
(iv)Describe a scenario where you might choose to use predominantly NOR gates in a design.
(v)What are some practical limitations or considerations when using universal gates like NAND or NOR gates?

Assignment3

(i)How can you implement an XNOR gate using basic logic gates
(ii)How can you implement an XOR gate using basic logic gates (AND, OR, NOT)?
(iii)What is the relationship between XOR and XNOR gates?
(iv)What is the purpose of an XOR gate?
(v)Can you provide an example of a real-world situation that can be represented using an XOR gate?

Assignment 4

(i)Why is a half adder insufficient for multi-bit addition?
(ii)Can you explain how a half adder contributes to larger arithmetic operations?
(iii)What happens if both inputs of a half adder are set to 1?
(iv)What's the key difference between a half adder and a full adder?

Assignment 5

(i)How can you chain multiple full adders to perform multi-bit addition?
(ii)What's the purpose of the carry input (Cin) in a full adder?
(iii)Can you simplify a full adder's logic equations?

Assignment 6

(i)Explain how a half subtractor can be used as part of a sequential logic circuit, providing a brief example
(ii)How does the power consumption of a half subtractor change with the input frequency?
(iii)How would the performance and characteristics of a half subtractor change if it were implemented using smaller transistors due to technology scaling in semiconductor manufacturing?

Assignment 7

(i)Given the inputs A and B, and the corresponding outputs of your circuit, explain how you would interpret the resulting 4-bit binary difference and borrow outputs in terms of the subtraction operation performed.
(ii) In your schematic, highlight the critical path for borrow propagation. Explain how the borrow signal is calculated and propagated through the series of full subtractors.

Assignment 8

(i) Describe a real-world application where a multiplexer could be used to optimize resource utilization.
(ii)How can a multiplexer be used to combine multiple lower-speed data streams into a higher-speed data stream
(iii)If you need to select one out of sixteen 8-bit data inputs, what kind of multiplexer would you use, and how many how many data lines and control lines would it have?

Assignment9

(i)How does a demultiplexer differ from a decoder?
(ii)Can a demultiplexer be used to perform logical operations like AND or OR?

Assignment 10

(i) Describe a real-world application where an encoder can be used to convert multiple input conditions into a a compact binary representation.
(ii) How can an encoder be used to encode information from multiple sensors into a binary representation?

Assignment 11

(i) How does it convert a binary input code into multiple output lines?
(ii)Explain how decoders are used in memory systems for address decoding.

Assignment 12

(i) Explain how you can detect overflow in the 4-bit parallel adder. What conditions lead to an overflow situation?
(ii) Describe how the carry-out (Cout) bit for each bit position is calculated in the adder circuit. How are the carry inputs for each bit position determined?

Assignment 13

(i)How is hysteresis used in comparators?
(ii)How do you connect a comparator to a microcontroller?
(iii)What is an open-drain comparator output?

Assignment 14

(i)What is ALU? How it performs arithmetic and logic operations?

Assignment 15

You are designing a memory system for a microcontroller-based application. The memory system uses a Random Access Memory (RAM) module to store data. The RAM module has a capacity of 8 kilobytes (KB) and is organized into 2,048 rows and 8 columns. Each row stores 8 bits (1 byte) of data.
(i)Explain the sequence of steps involved in performing a read operation from the RAM module. Describe how the microcontroller can retrieve a specific byte of data from a given memory location.
(ii)Describe the process of performing a write operation to the RAM module. Explain how the microcontroller can store a specific byte of data into a designated memory location.

Extension and Follow-up Activities (if applicable)

(i)Generate D.C triangular waveform using microprocessor.
(ii)Design Digital Clock using microprocessor
(iii)Generate Sine wave using microprocessor
(iv)Design Stepper Motor using microprocessor
(v)Software for detection of zero crossing detector of an AC Waveform.

Assessments

As per the assessment and evaluation policy of University.

Suggested readings

1. Modern Digital Electronics R.P. Jain, McGraw-Hill, 4th Edition, 2009.
2. Digital Systems Tocci, Widmer and Moss, Pearson 12th Edition, 2017.

Programme Name and Semester: MCA 1st Semester
Course Name (Course Code): Computer Organization and Operating System (MCA17403)
Academic Session: 2025-26

Assignment copy format:

1. Write in A4 page; No channel file will be accepted.
2. Front page and Index page format will be provided by department
3. Following points must be included while writing assignment copy
    - Problem Definition
    - Pin Diagram
    - Circuit Diagram
    - Result
    - Discussion