Shreyas Sawant
D7A, 55
DOP - 24-09-2021
DOS - 11-10-2021

DLCOA Assignment No. 1

1. Convert $(58.34)_{10}$ into binary, hexadecimal and octal number systems.

Ans $(58.34)_{10} = (x)_2$

	Quotient	Remainder	
58/2	29	0	
29/2	14	1	
14/2	7	0	
7/2	3	1	
3/2	1	1	↑
1/2	0	1	

$(58)_{10} = (111010)_2$

Now, $(0.34)_{10} = 0.34 \times 2 = 0.68$ 0 ↓

$= 0.68 \times 2 = 1.36$ 1

$= 0.36 \times 2 = 0.72$ 0

$= 0.72 \times 2 = 1.44$ 1

$(0.34)_{10} = (0.0101)_2$

$\therefore (58.34)_{10} = (111010.0101)_2$

$(58.34)_{10} = (x)_{16}$

	Quotient	Remainder	
58/16	3	10 = A	↑
3/16	0	3	

$(58)_{10} = (3A)_{16}$

Now, $(0.34)_{10}$ = 0.34 × 16 = 5.44 = 5

= 0.44 × 16 = 7.04 = 7

= 0.04 × 16 = 0.64 = 0

= 0.64 × 16 = 10.24 = 10 = A

$(0.34)_{10} = (0.570A)_{16}$

$\therefore (58.34)_{10} = (3A.570A)_{16}$

$(58.34)_{10} = (x)_8$

	Quotient	Remainder
58/8	7	2
7/8	0	7

$(58)_{10} = (72)_8$

Now, $(0.34)_{10}$ = 0.34 × 8 = 2.72 = 2

= 0.72 × 8 = 5.76 = 5

= 0.76 × 8 = 6.08 = 6

= 0.08 × 8 = 0.64 = 0

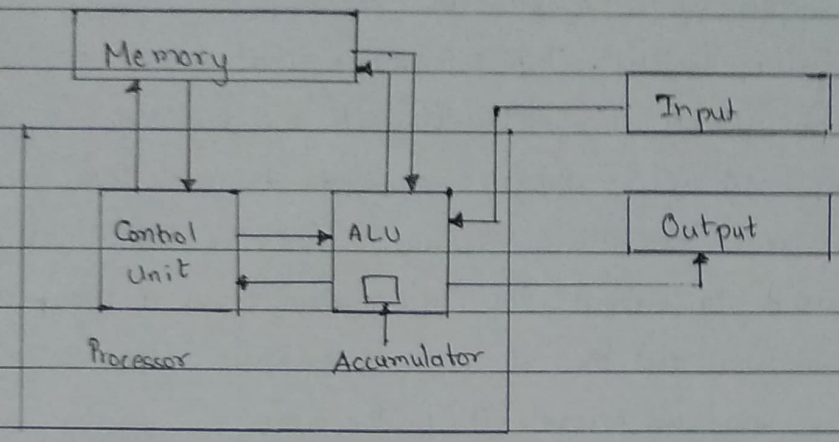
$(0.34)_{10} = (0.2560)_8$

$\therefore (58.34)_{10} = (72.2560)_8$

2. Explain Von Neumann Architecture

Ans The modern computers are based on stored program concept introduced by John Von Neumann. In this stored-program concept programs and data are stored in a seperate storage units called memories and are treated the same. This novel idea meant that a computer built with this architecture would be much easier to reprogram.

The basic structure of Von-Neumann architecture is like:



It is also known as IAS computer and is having three basic units:-

(i) The central processing units (CPU)
(ii) The main memory unit
(iii) The input/output device

Some basic components discussed in details:

(A) CONTROL UNIT : A control unit handles all processor control signals. It directs all input and output flow, fetches code for instructions and controlling how data moves around the system.

(B) ARITHEMATIC & LOGIC UNIT (ALU): It is a that part of the CPU that handles all calculations the CPU may need, eg - addition, subtraction, comparisons. It performs logical operations, bit shifting operations and arithematic operation

(C) MAIN MEMORY UNIT : Main memory consists of:-

- Accumulator - Stores the results of calculations made by ALU
- Program counter - Keeps track of memory location of next instructions to be dealt with.
- Memory Address Register - Stores memory locations of instructions that need to be fetched from memory.
- Memory Data Register - Stores the instructions fetched from memory or any data that is to be transferred to or stored in memory.

- Current Instruction Register : Stores the most recently fetched instructions while it is waiting to be coded and executed.
- Instruction Buffer Register: The instruction that is not to be executed immediately is placed in instruction buffer register.

D) INPUT/OUTPUT DEVICES: Program or data is read into main memory from input device or secondary storage under the control of CPU. Output devices are used to output the information from a computer. If some results are evaluated and then stored in a computer, then with help of output devices, we can present it to the user.

E) BUSES: Data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory.

Types of buses :-

- Data Bus : It carries data among memory unit, I/O devices and the processor.
- Address Bus : It carries the address of data between memory and processor.
- Control Bus : It carries control commands from the CPU in order to control and co-ordinate all the activities within the computer.

This architecture is very important and is used in our PCs and even in super computers.

3. Explain the EXCESS-3 and ASCII codes

Ans EXCESS-3

This is another form of BCD code, on which each decimal digit is coded into 4 bit binary code. It is non-weighted code.

The code for each decimal digit is obtained by adding 3 to natural BCD code of the digit.

The code is self-complementing code, which means 1's complement of the coded number yields 9's complement of number itself

Example -

BCD of 2 is - 0010

Excess - 3 code of 2 is given by -

$$\begin{array}{r} 0010 \\ +\ 0011 \\ \hline 0101 \end{array}$$

The self complementing property of this code is used in performing subtraction operation in digital systems.

ASCII Code :-

ASCII stands for American Standard Code for Information Interchange

It is universally accepted alphanumeric code.

It is 7-bit code in which the decimal digits are represented by the BCD code preceded by 011. Since it is 7 bit code, it represent $2^7 = 128$ characters

ASCII characters can be broadly divided into :-

(i) Control characters (0 - 31 and 127)

(ii) Printable characters (32 - 126)

(iii) extended ASCII characters (128 - 255)

A has an ASCII code of 65

Z has an ASCII code of 90

a has an ASCII code of 97

z has an ASCII code of 122

Spcare character has an ASCII code of 32.

After 127 ie 128-255 we have extended ASCII representing mathematical and other symbols, that are not represented as keys and are not used in general.

4. Draw block diagram of half adder, full adder, 4:1 multiplexer and 3:8 decoder with truth tables

Ans HALF ADDER

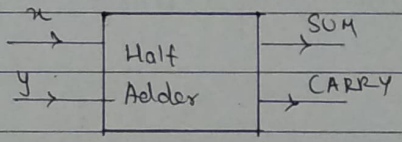
A combinational circuit that performs the addition of two bits is called half adder.

The circuit takes two binary inputs and two binary outputs.

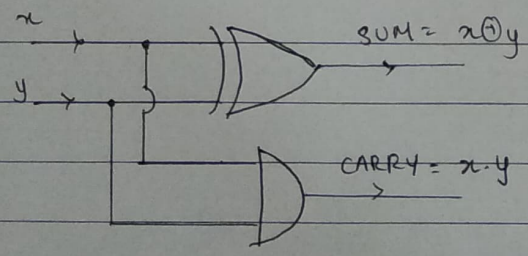
Truth table:

x	y	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Block diagram:



Logic Gate Diagram:



FULL ADDER

A combinational circuit that performs addition of three bits is a full adder. It consists of three inputs and two outputs.

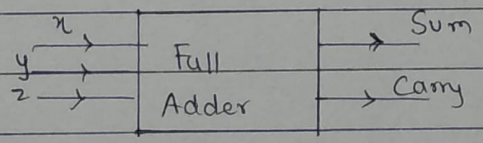
Truth table

x	y	z	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

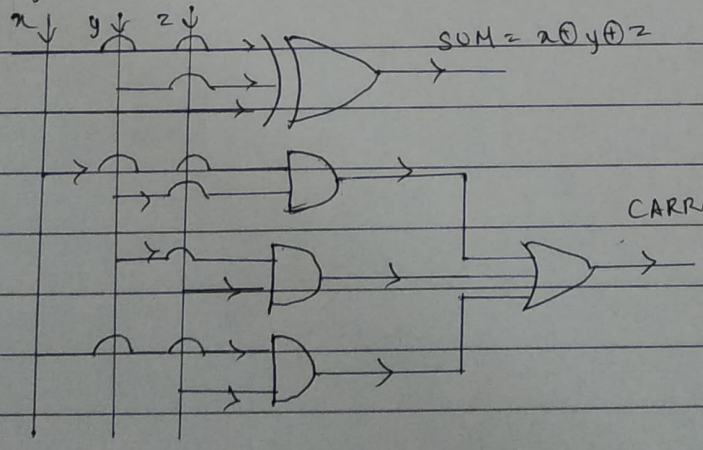
$SUM = x \oplus y \oplus z$

$CARRY = xy + yz + xz$

Block Diagram



Logic Gate Diagram



4:1 Multiplexer

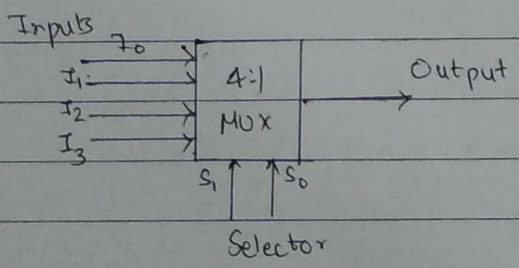
A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input lines and n-selection lines whose bit combination determine which input is selected.

A 4:1 multiplexer contains of four inputs I_0 to I_3 applied to one input of AND Gate. Selection lines S_1 and S_0 are decoded to select a particular AND Gate. The output are applied to a single OR gate that provides one line input.

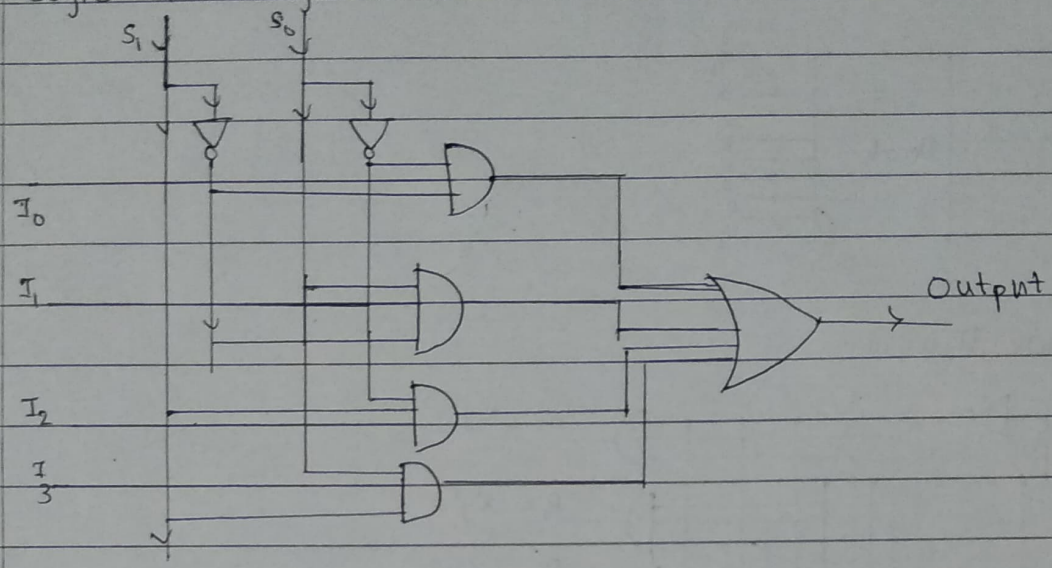
Truth table:

S_1	S_0	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Block Diagram



Logic Gate Diagram



3:8 DECODER

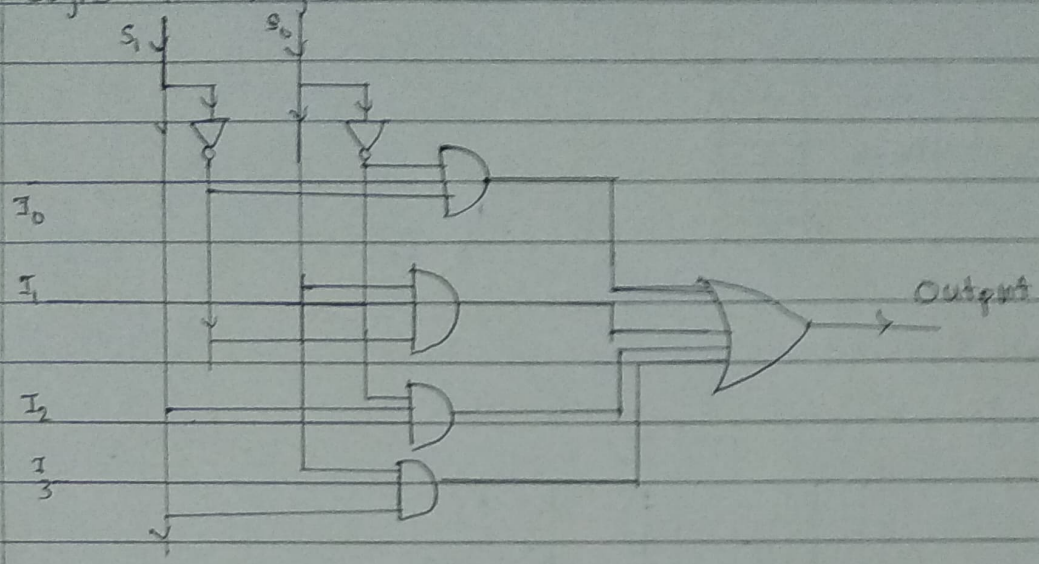
A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

In 3:8 line decoder, the three inputs are decoded into eight outputs

Truth table:

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Logic Gate Diagram



3:8 DECODER

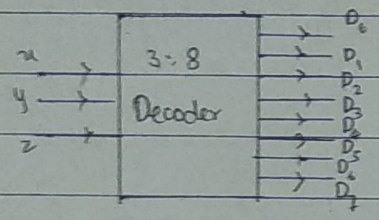
A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

In 3:8 line decoder, the three inputs are decoded into eight outputs.

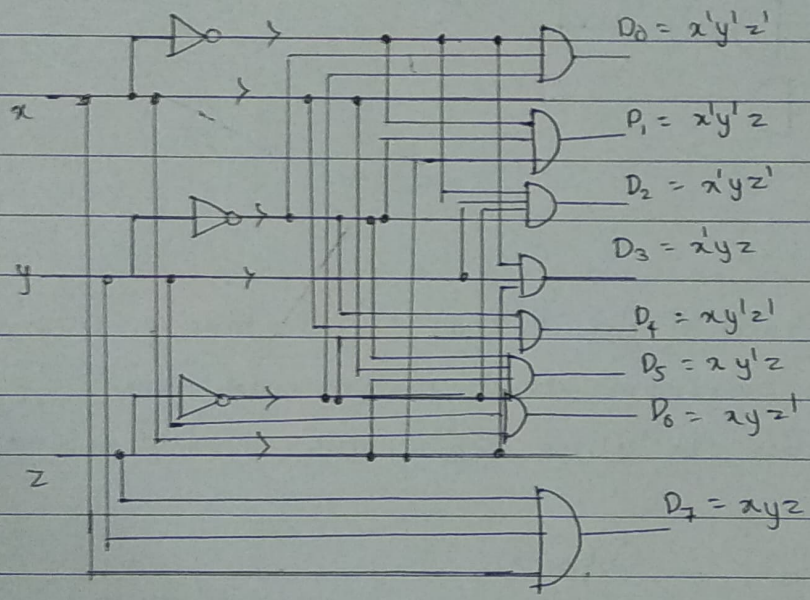
Truth table:

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Block diagram:



Logic Gate Diagram



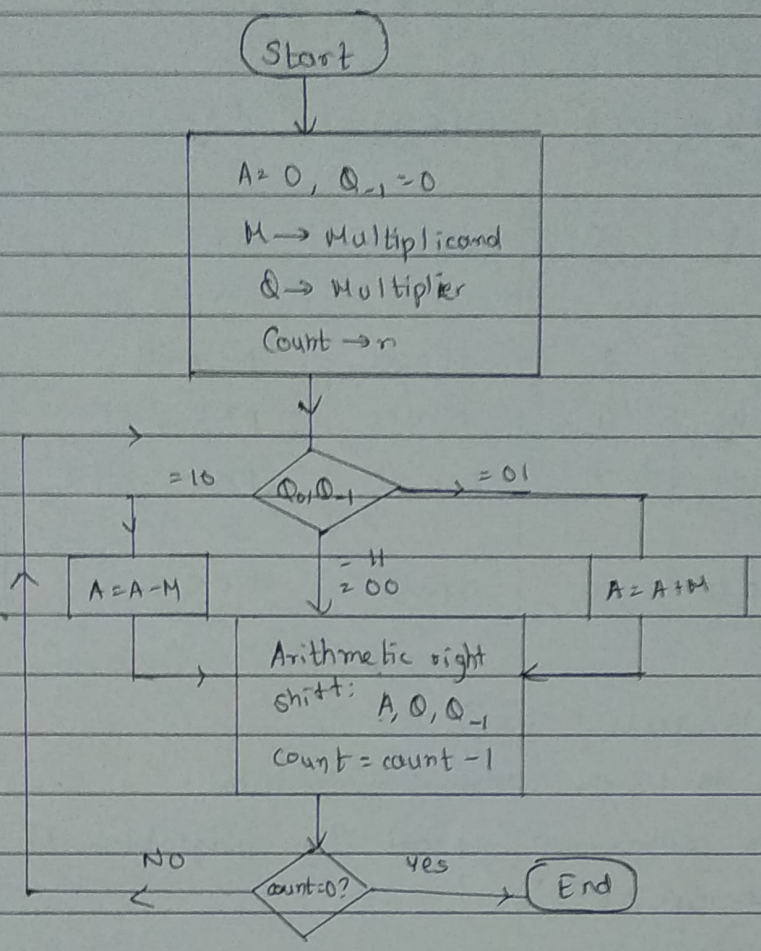
5. Draw the flowchart of Booth's Algorithm and solve the following

M = 8 Q = -4

Ans M = 8, Q = -4

A = 0

FLOWCHART OF BOOTH'S ALGORITHM -



Given : $M = 8 = (01000)_2$

$Q = -4 = (11100)_2$

$-M = -8 = (11000)_2$

n = no. of bits = 5

	A	Q	Q_{-1}	M	Status
n=5	00000	11100	0	01000	Initialization
n=4	00000	01110	0	01000	ARS
n=3	00000	00111	0	01000	ARS
n=2	11000	00111	0	01000	A=A-M
	11100	00011	1	01000	ARS
n=1	11110	00001	1	01000	ARS
n=0	11111	00000	1	01000	ARS

STOP

The answer is $= (11111\ 00000)_2$

Since the MSB is 1, the answer is negative

So we take 2's complement to get the magnitude of the number

$8 \times -4 = -(00001\ 00000)_2$

$8 \times -4 = (-32)_{10}$

6. Explain the restoring division method with an example.

Ans A division algorithm provides a quotient and a remainder when we divide two numbers. Restoring division algorithm is a shown division algorithm. The algorithm is called restoring because the value in register A is restored after each iteration.

Register Q contains quotient and A contains remainder. The n-bit dividend is loaded in Q and divisor is loaded in M.

Value of register is kept 0 and this is the register whose value is restored during iterations.

Algorithm :-

Step 1 : First the registers are initialized with the corresponding values.

Step 2 : Then the content of register A and Q is shifted left as if they are a single unit.

Step 3 : Then content of register M is subtracted from A and result stored in A

Step 4 : Then the most significant bit of the A is checked if it is 0 the least significant bit of Q is set to 1 otherwise, if it is 1 the LSB of Q is set to 0 and the value of A is restored i.e. the value of A before sub.

Step 5 : The value of counter n is decremented.

Step 6 : If the value of n becomes zero, we get out of loop, else repeat step 2.

Step 7 : Finally the register Q contains the quotient and A contains the remainder.

Example :-

Dividend = 11

Divisor = 3

Q = Dividend = 11 = $(1011)_2$, unsigned

M = Divisor = 3 = $(0011)_2$ $\quad -M = (1101)_2$

n = no. of bits = 4

A = $(00000)_2$

	A	Q	M	Status
n=4	0 0000	1011	00011	Initialization
	00001	011_	00011	left shift AQ
	11110	011_	00011	A = A-M
	00001	0110	00011	MSB[A] = 1 $\Rightarrow Q_0 = 0$, restore A

	A	Q	M	Status
n=3	00010	110_	00011	Left shift AQ
	11111	110_	00011	A = A-M
	00010	1100	00011	MSB[A]=1 ⇒ Q_0=0, restore A
n=2	00101	100_	00011	Left shift AQ
	00010	100_	00011	A = A-M
	00010	1001	00011	MSB[A]=0 ⇒ Q_0=1
n=1	00101	001_	00011	Left shift AQ
	00010	001_	00011	A = A-M
	00010	0011	00011	MSB[A]=0 ⇒ Q_0=1

n=0
STOP

Q = Quotient = $(0011)_2$ = $(3)_{10}$

A = Remainder = $(00010)_2$ = $(2)_{10}$

∴ 11/3 remainder = 2, quotient = 3

7. Explain different addressing modes of processor.

Ans The term addressing modes refer to the way in which the operand of an instruction is specified. The addressing mode specifies a rule interpreting or modifying the address field of the instruction before the operand is actually executed.

(i) Immediate addressing mode - In immediate addressing mode, the source operand is always data. If data is 8-bit, then the instruction will be of 2 bytes. Eg - MVI B 45

(ii) Register addressing mode - In register addressing mode, the data to be operated is available inside the registers and registers are operands.

Eg - MOV A,B ; INRA

(iii) Direct addressing mode: The data to be operated is available inside a memory location and that location is specified as an operand.

eg - LDA 2050

(iv) Register indirect addressing mode : In this mode, the data to be operated is available inside a memory location is indirectly specified by a register pair.

(v) Implied / implicit addressing mode: In this addressing mode, the operand is hidden and the data to be operated is available in the instruction itself. eg - CMA ; RRC.