ASSIGNMENT NO. 01

TTILE: To study the Realization of Full Adder and Subtractor using
a) Basic Gates and b) Universal Gates

PROBLEM STATEMENT: To Realize Full Adder and Subtractor using

a) Basic Gates and b) Universal Gates

PRE-REQUISITE:

Digital trainer kit, ICs- 74LS08, 74LS86, 74LS00, 74LS02, Probs

THEORY:

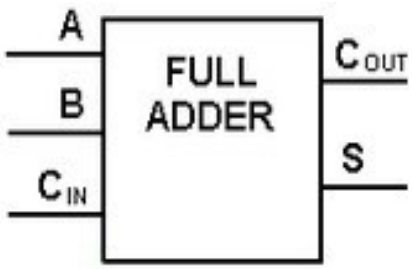
1) Implementation of Full Adder:

A full adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and two outputs. Two of these variables denoted by A and B represent the two significant bits to be added. The third input represents the carry from previous lower significant position. From the truth-table, the full adder logic can be implemented. We can see that the output S is an EXOR between the input A and the half-adder SUM output with B and CIN inputs. We must also note that the COUT will only be true if any of the two inputs out of the three are HIGH.

Thus, we can implement a full adder circuit with the help of two half adder circuits. The first will half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add CIN to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus, COUT will be an OR function of the half-adder Carry outputs. Take a look at the Logic Diagram & implementation of the full adder circuit shown below.

Implementation of Full Adder using Basic Gates:

Logic Diagram:

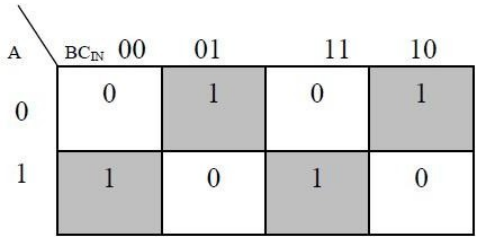


Truth Table for Design of full adder:

Input			*Output*	
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

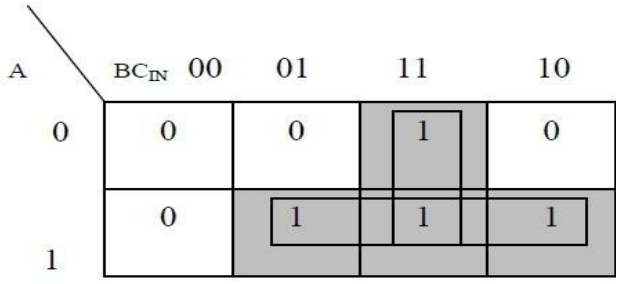
Based on the truth table, the Boolean functions for Sum (S) and Carry – out (COUT) can be derived using

K – Map For Sum S



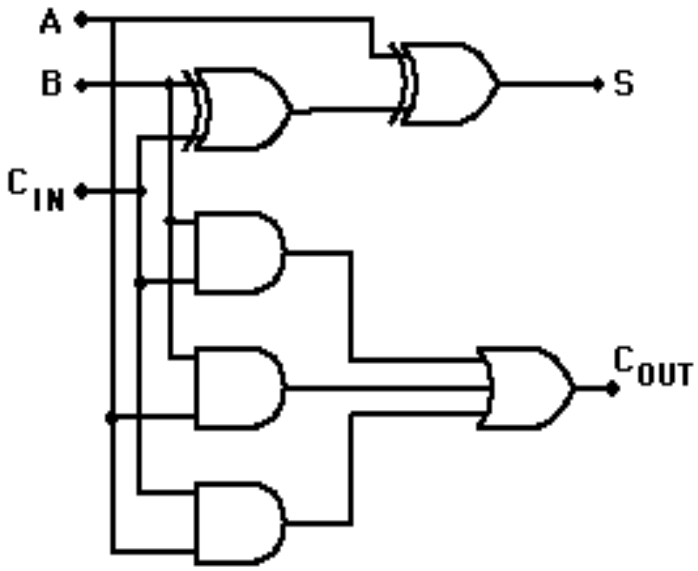
SUM = A EXOR B EXOR Cin

K – Map For Carry C



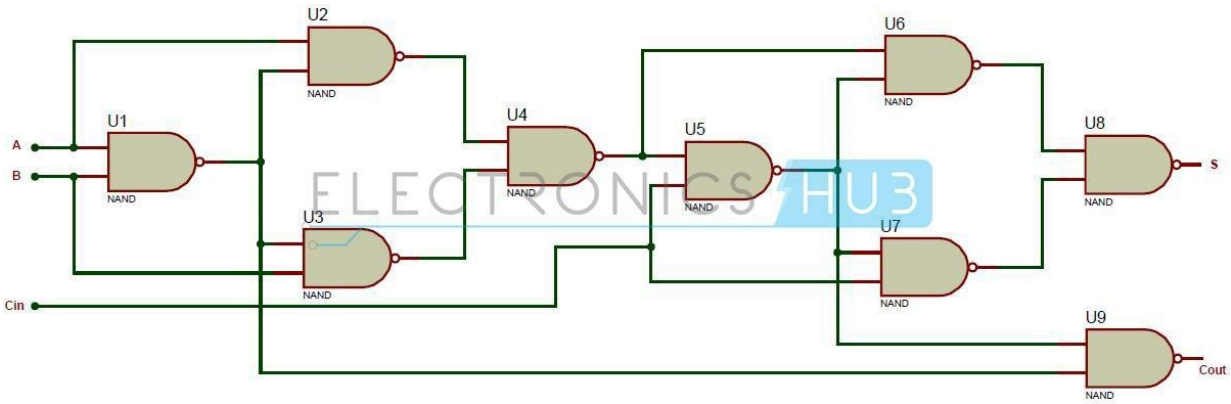
1. **Carry = (A AND B) OR (A AND C) OR (B AND C)**

Logic Circuit for Full Adder



Implementation of Full Adder using Universal Gates:

As mentioned earlier, a NAND gate is one of the universal gates and can be used to implement any logic design. The circuit of full adder using only NAND gates is shown below.



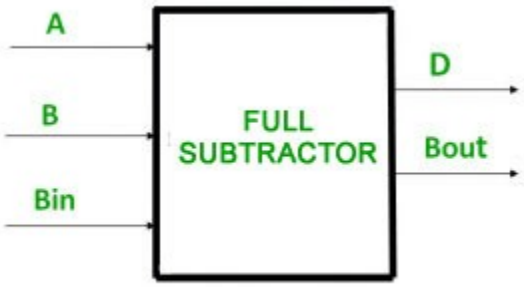
2) Implementation of Full Subtractor:

The full subtractor is a Combinational Circuit which is used to perform subtraction of three input bits : the minuend , subtrahend , and borrow in . The full subtractor generates two output bits: the difference and borrow out . is set when the previous digit borrowed from . Thus, is also subtracted from as well as the subtrahend. Like the half subtractor, the full subtractor generates a borrow out when it needs to borrow from the next digit. Since we are subtracting by and , a borrow out needs to be generated when .

When a borrow out is generated, 2 is added in the current digit. (This is similar to the subtraction algorithm in decimal. Instead of adding 2, we add 10 when borrow.)

Implementation of Full Subtravtor using Basic Gates:

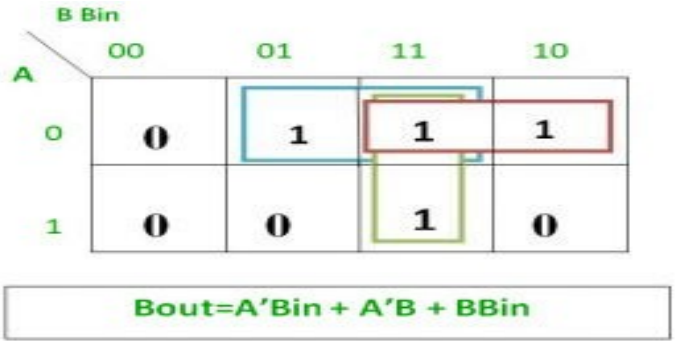
Logic Diagram:



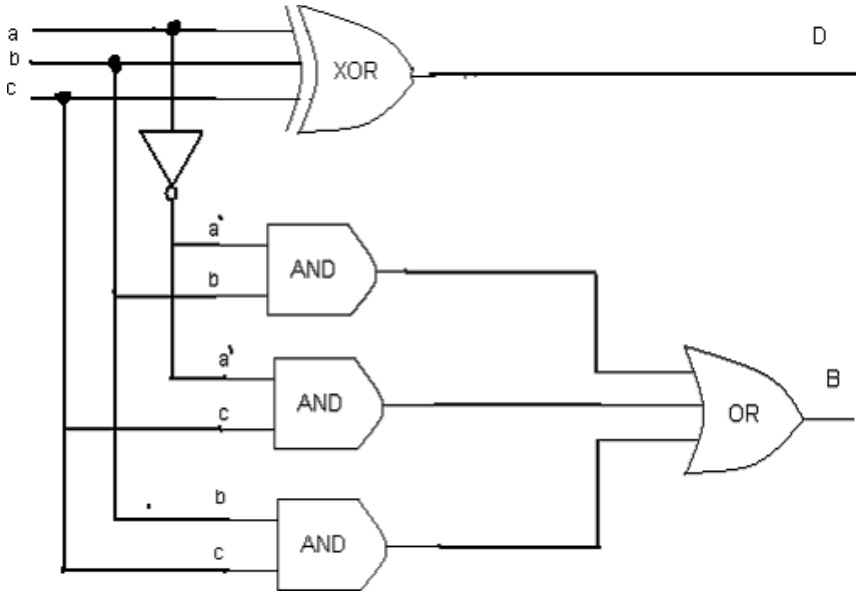
INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

From above table we can draw the K-Map as shown for “difference” and “borrow”.

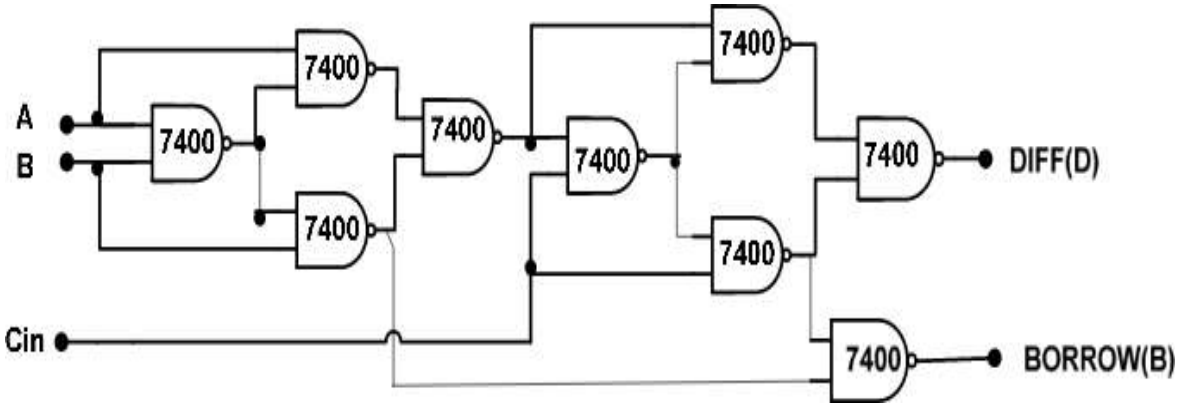
Difference= A EXOR B EXOR Bin



Logic Circuit for Full Subtractor



2. Implementation of Full Subtractor using Universal Gates:



QUESTIONS:

1. Which are universal gates?
2. What NAND gate and NOR gate called Universal gates?
3. Explain types of Ic's.
4. What is Full Adder?
5. Disadvantage of Half Adder.
6. What is Full subtractor?
7. Disadvantage of Half subtractor?
8. How Full adder is implemented using half adder?

ASSIGNMENT NO. 02

TITLE: Code Converters e.g. BCD to Excess-3 & Binary to Gray

PROBLEM STATEMENT: To study the design & implementation of

a) Binary to Gray code converter using logic gates
b) BCD to Excess-3 code converter using logic gates

PRE-REQUESITE:
Digital trainer kit, ICs- 74LS08, 74LS86, 74LS00, 74LS02, 7432, 7404 , Probs

THEORY:

There is a wide variety of binary codes used in digital systems. Some of these codes are binary-coded -decimal (BCD), Excess-3, Gray, octal, hexadecimal, etc. Often it is required to convert from one code to another. For example the input to a digital system may be in natural BCD and output may be 7-segment LEDs. The digital system used may be capable of processing the data in straight binary format. Therefore, the data has to be converted from one type of code to another type for different purpose. The various code converters can be designed using gates.

1) Binary Code:

It is straight binary code. The binary number system (with base 2) represents values using two symbols, typically 0 and 1.Computers call these bits as either off (0) or on (1). The binary code are made up of only zeros and ones, and used in computers to stand for letters and digits. It is used to represent numbers using natural or straight binary form.It is a weighted code since a weight is assigned to every position. Various arithmetic operations can be performed in this form. Binary code is weighted and sequential code.

2) Gray Code:

It is a modified binary code in which a decimal number is represented in binary form in such a way that each Gray- Code number differs from the preceding and the succeeding number by a single bit. (E.g. for decimal number 5 the equivalent Gray code is 0111 and for 6 it is 0101. These two codes differ by only one bit position i. e. third from the left.) Whereas by using binary code there is a possibility of change of all bits if we move from one number to other in sequence (e.g. binary code for 7 is 0111 and for 8 it is 1000). Therefore it is more useful to use Gray code in some applications than binary code.The Gray code is a non weighted code i.e. there are no specific weights assigned to the bit positions. Like binary numbers, the Gray code can have any no. of bits. It is also known as reflected code.

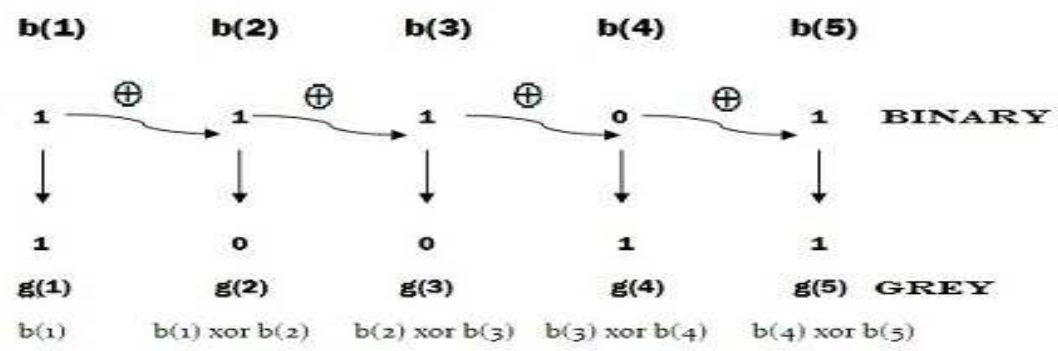
Applications:

1. Important feature of Gray code is it exhibits only a single bit change from one code word to the next in sequence. This property is important in many applications such as Shaft encoders where error susceptibility increases with number of bit changes between adjacent numbers in sequence.
2. It is sometimes convenient to use the Gray code to represent the digital data converted from the analog data (Outputs of ADC).
3. Gray codes are used in angle-measuring devices in preference to straight forward binary encoding.
4. Gray codes are widely used in K-map.

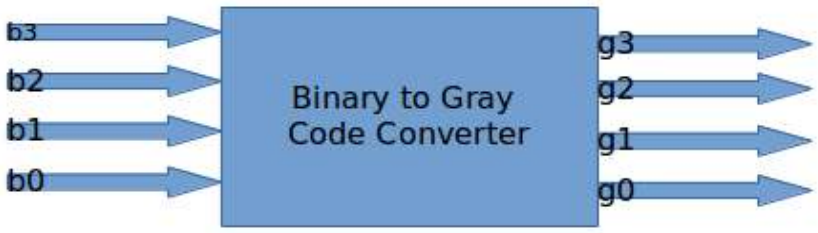
The disadvantage of Gray code is that it is not good for arithmetic operation

Binary To Gray Conversion

1. Record the most significant bit as it is.
2. EX-OR this bit to the next position bit, record the resultant bit.
3. Record successive EX-ORed bits until completed.
4. Convert 0011 binary to Gray.



Block Diagram:

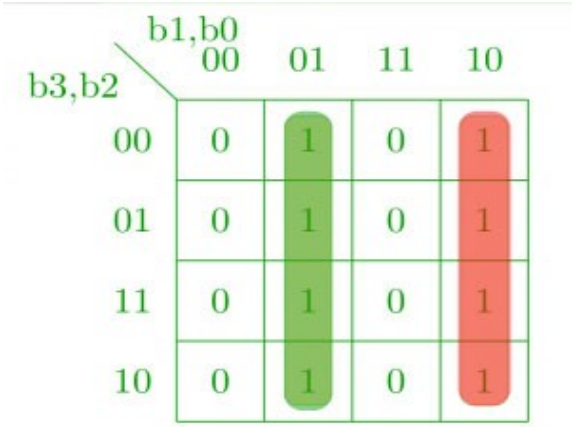


Truth Table:

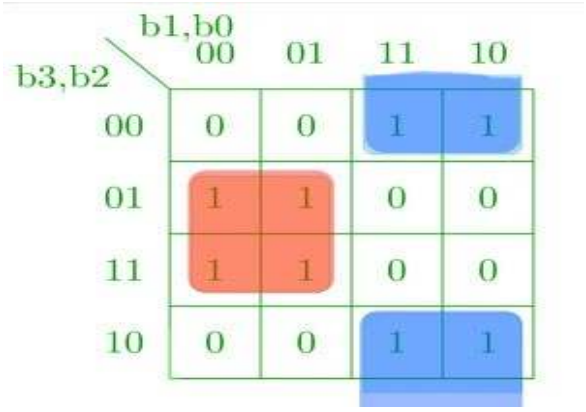
Binary				Gray Code			
b_3	b_2	b_1	b_0	g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

K-map

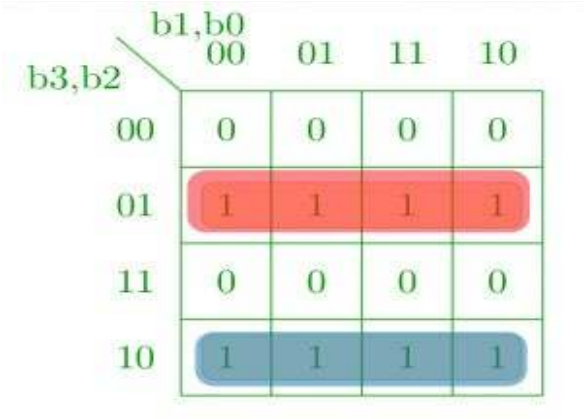
K-map for g0:



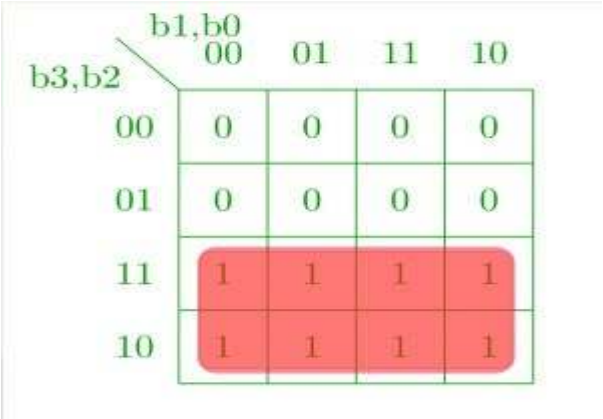
K-map for g1:



K-map for g2:



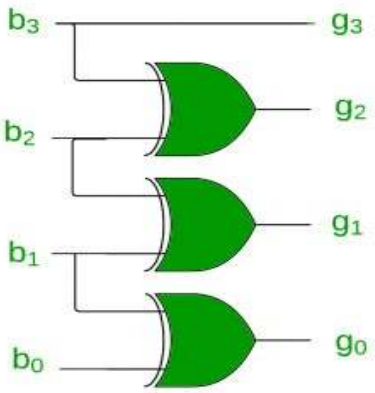
K-map for g3:



Corresponding minimized boolean expressions for gray code bits –

$$g_0 = b_0b_1' + b_1b_0' = b_0 \oplus b_1$$
$$g_1 = b_2b_1' + b_1b_2' = b_1 \oplus b_2$$
$$g_2 = b_2b_3' + b_3b_2' = b_2 \oplus b_3$$
$$g_3 = b_3$$

Digital circuit



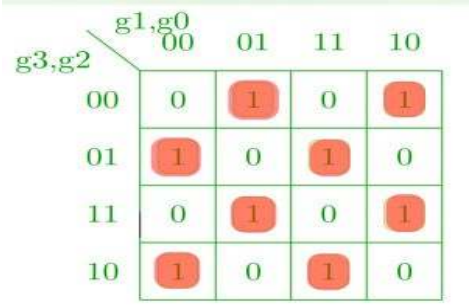
Gray to Binary Code Conversion:

Truth Table:

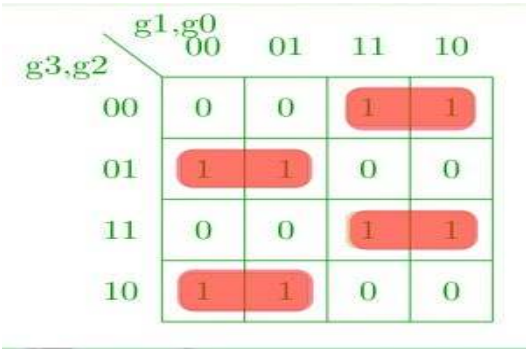
Gray Code				Binary			
g_3	g_2	g_1	g_0	b_3	b_2	b_1	b_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

Kmap:

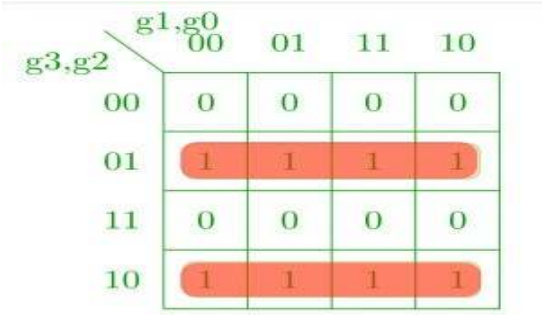
K-map for b0:



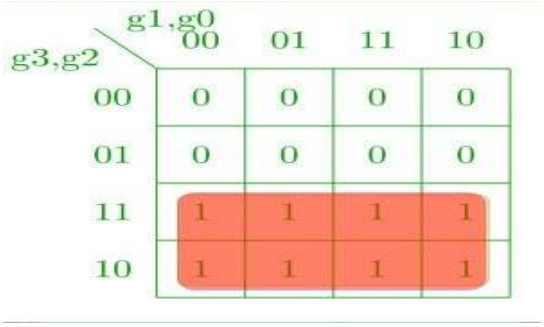
K-map for b1:



K-map for b2:



K-map for b3:



Corresponding Boolean expressions

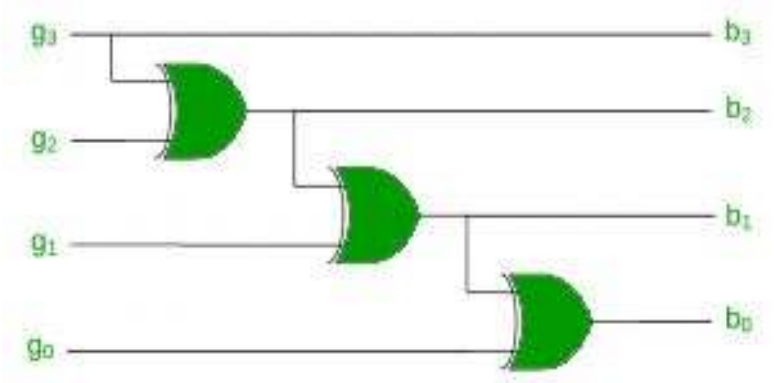
$B3=G3$

$B2=G3 \oplus G2$

$B1=G3 \oplus G2 \oplus G1$

$B0=G3 \oplus G2 \oplus G1 \oplus G0$

Digital circuit



3) BCD Code:

Binary Coded Decimal (BCD) is used to represent each of decimal digits (0 to 9) with a 4-bit binary code. For example $(23)_{10}$ is represented by 0010 0011 using BCD code rather

than$(10111)_2$ This code is also known as 8-4-2-1 code as 8421 indicates the binary weights of four bits(2^3, 2^2, 2^1, 2^0). It is easy to convert between BCD code numbers and the familiar decimal numbers. It is the main advantage of this code. With four bits, sixteen numbers (0000 to 1111) can be represented, but in BCD code only 10 of these are used. The six code combinations (1010 to 1111) are not used and are invalid.

Applications: Some early computers processed BCD numbers. Arithmetic operations can be performed using this code. Input to a digital system may be in natural BCD and output may be 7-segment LEDs. It is observed that more number of bits are required to code a decimal number using BCD code than using the straight binary code. However in spite of this disadvantage it is very convenient and useful code for input and output operations in digital systems.

Fig. 3 BCD Coded Decimal Representation

4) EXCESS-3 Code:

Excess-3, also called XS3, is a non weighted code used to express decimal numbers. It can be used for the representation of multi-digit decimal numbers as can BCD.The code for each decimal number is obtained by adding decimal 3 and then converting it to a 4-bit binary number. For e.g. decimal 2 is coded as 0010 + 0011 = 0101 in Excess-3 code.

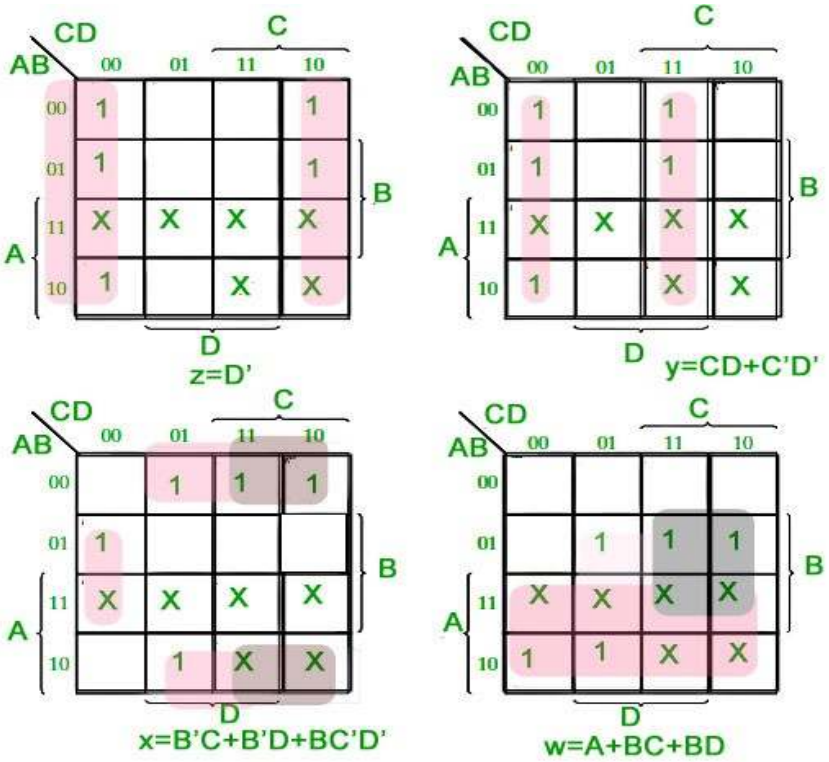
This is self complementing code which means 1's complement of the coded number yields 9's complement of the number itself. Self complementing property of this helps considerably in performing subtraction operation in digital systems, so this code is used for certain arithmetic operations.

BCD to Excess-3 Code Conversion

Truth Table:

BCD(8421)				Excess-3			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

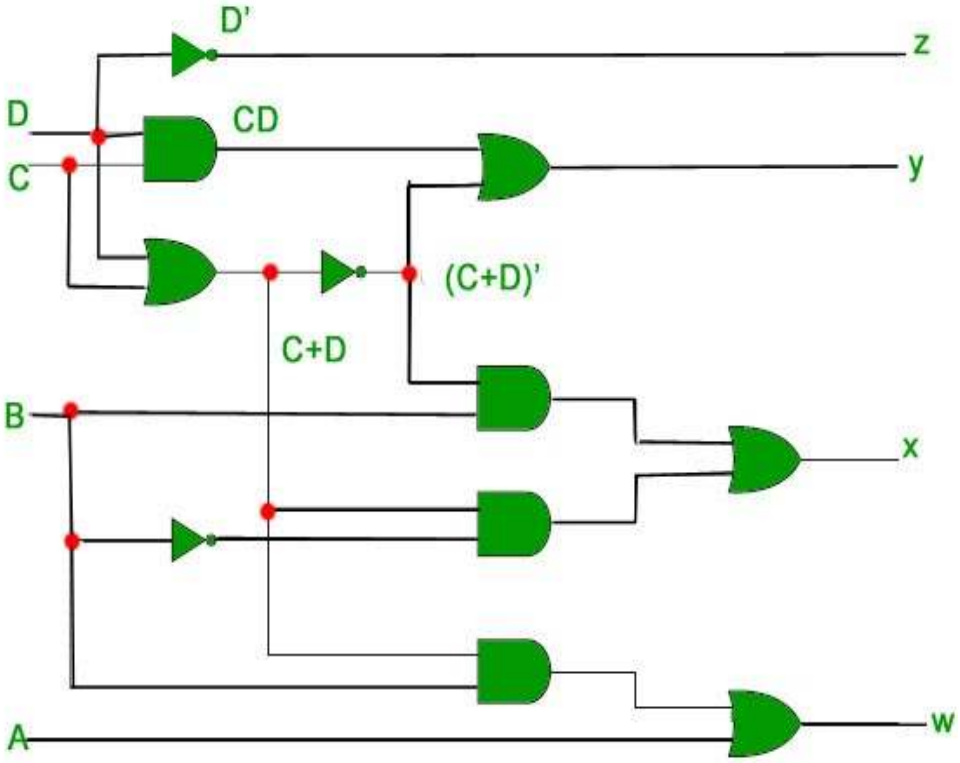
Kmap



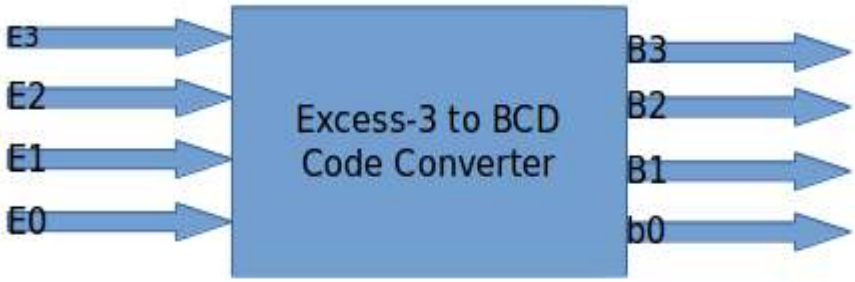
Corresponding Boolean expressions:

$$w = A + BC + BD$$
$$x = B'C + B'D + BC'D'$$
$$y = CD + C'D'$$
$$z = D'$$

Digital Circuit



Excess-3 to BCD code Conversion

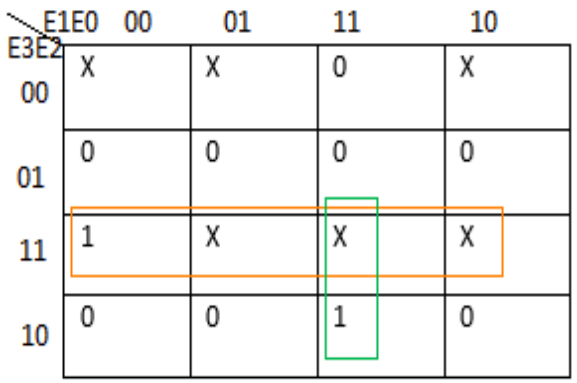


Truth Table:

EXCESS-3 INPUT				BCD OUTPUT			
E3	E2	E1	E0	B3	B2	B1	B0
0	0	0	0	X	X	X	X
0	0	0	1	X	X	X	X
0	0	1	0	X	X	X	X
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

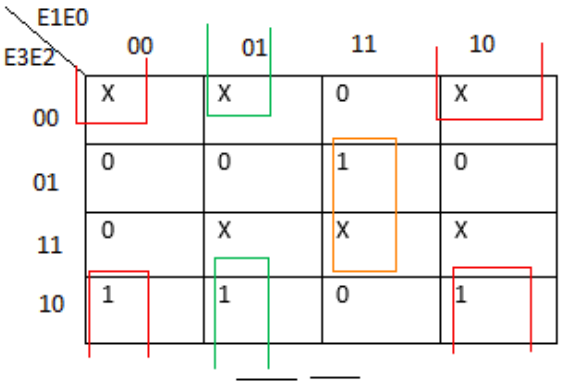
Kmap

K-map for B3:



B3 = ((E0E1) + E2) E3

K-map for B2:



$B2 = \overline{E2}E1 + \overline{E2}E0 + E2E1E0$

K-map for B1:

E3E2 \ E1E0	00	01	11	10
00	X	X	0	X
01	0	1	X	1
11	0	X	X	X
10	X	1	0	1

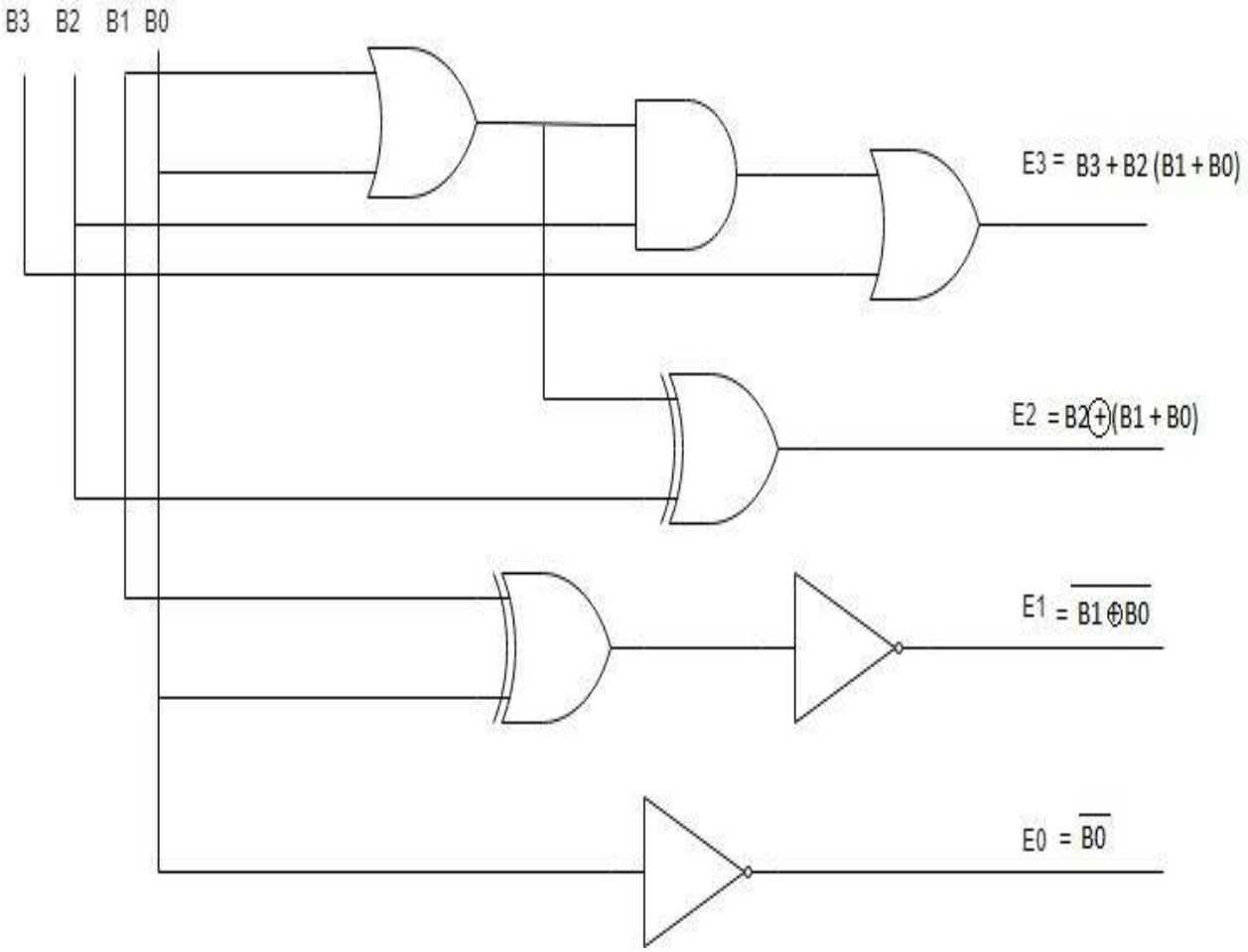
B1 = E1 ⊕ E0

Kmap for B0:

E3E2 \ E1E0	00	01	11	10
00	X	X	0	X
01	1	0	0	1
11	1	X	X	X
10	1	0	0	1

$B0 = \overline{E0}$

Digital circuit



QUESTIONS:

1. What is difference between binary to BCD number syatem?
2. What are properties of Gray code?
3. What are properties of Excess-3 code?
4. What is non-weighted code?
5. Whay gray code called unit distance code?
6. Applications of Gray code.

ASSIGNMENT NO: 03

AIM: Application like Realization of Boolean expression using MUX / DEMUX.

PROBLEM STATEMENT: Realization of Boolean expression for suitable combination logic using MUX 74151.

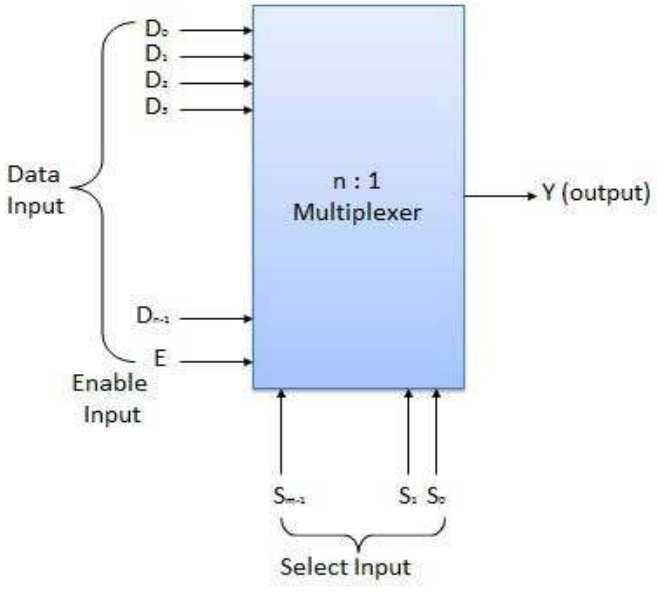
PRE-REQUESITE:

Digital trainer kit, ICs- 74151, 74LS08, 7404 Probs

THEORY:

Multiplexer (MUX)

Multiplexer means transmitting a large number of information units over a smaller number of channels or lines. A digital 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 line and n selection lines whose bit combination determine which input is selected.



Neccessity of Multiplexer:

Multiplexing technique is designed to reduce the number of electrical connections or leads in the display matrix. Whereas driving signals are applied not to each pixel (picture element) individually but to a group of rows and columns at a time. Besides reducing the number of individually independent interconnections, multiplexing also simplifies the drive electronics,

reduces the cost and provides direct interface with the microprocessors. There are limitations in multiplexing due to complex electro-optical response of the liquid crystal cell. However, fairly reasonable level of multiplexing can be achieved by properly choosing the multiplexing scheme, liquid crystal mixture and cell designing.

Types of Multiplexer:

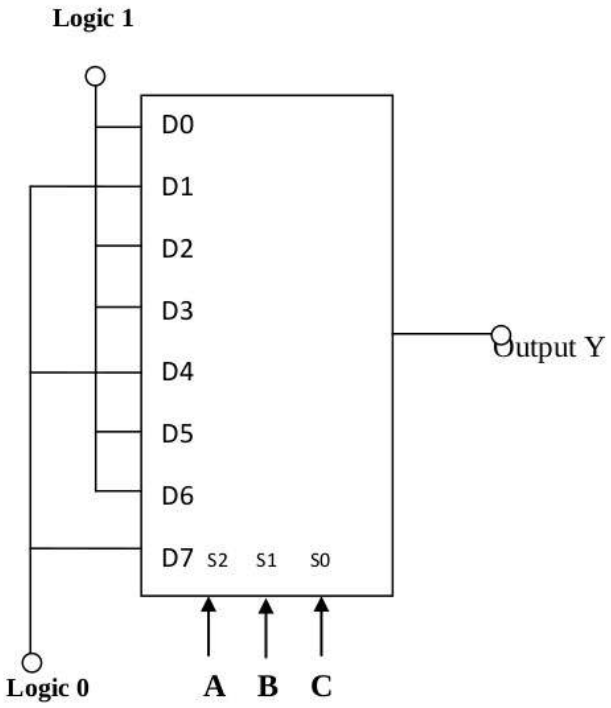
- 2-1 multiplexer (1select line)
- 4-1 multiplexer (2 select lines)
- 8-1 multiplexer(3 select lines)
- 16-1 multiplexer (4 select lines)

1. **Implement the following expression using a MUX F(A,B,C) = ∑ m(0,2,3,5,6)**

Solution:

As there are three variables A,B and C, the MUX is having 3 select inputs. The function ∑ m gives numbers corresponding to each minterm as (0,2,3,5,6).

Thus, implementation of logic expression using a MUX is as shown below:



2. Implement the following Boolean function using 8:1

MUX F(A,B,C,D) = ∑ m(2,5,7,8,9,10,13,15)

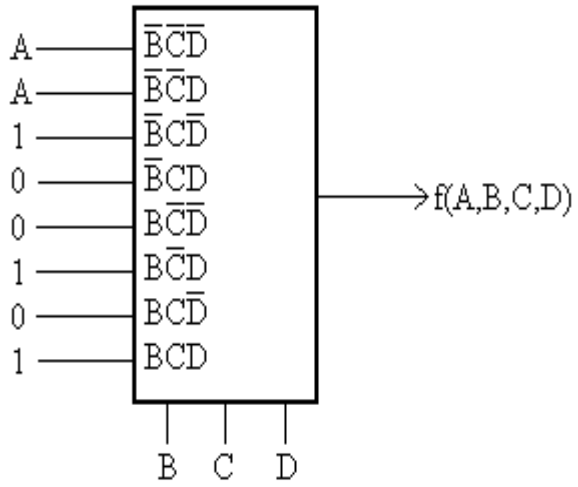
Solution:

The variables B,C,D are connected to the select lines S2, S1 & S0 respectively & A take as extra select input.

ll the minterms are then listed together with the data variable set as true and false in order to find the data input values.

4 2 1	0 8	
B C D	$\overline{A}$ A	Data Value
0 0 0	0 ⑧	A
0 0 1	1 ⑨	A
0 1 0	② ⑩	1
0 1 1	3 11	0
1 0 0	4 12	0
1 0 1	⑤ ⑬	1
1 1 0	6 14	0
1 1 1	⑦ ⑮	1

Thus, implementation of logic expression using a MUX is as shown below:



Applications of Multiplexers:

1. Communication System
2. Computer Memory
3. Telephone Network

4. Transmission from the Computer System of a Satellite

QUESTIONS:

1. **What** is multiplexer?
2. Why MUX called data distributor?
3. What is differnce between MUX and DEMUX?
4. What is Enable(E)?
5. What are apllications of MUX?

ASSIGNMENT NO: 04

TITLE: Magnitude Comparator

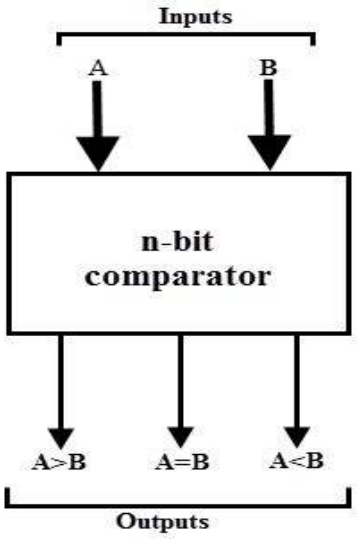
PROBLEM STATEMENT: To Verify the truth table of one bit and two bit comparators using logic gates and four bit and eight bit comparators using IC 7485.

PRE-REQUESITE:

Digital trainer kit, ICs- 7485, 74LS08, 7404 Probs

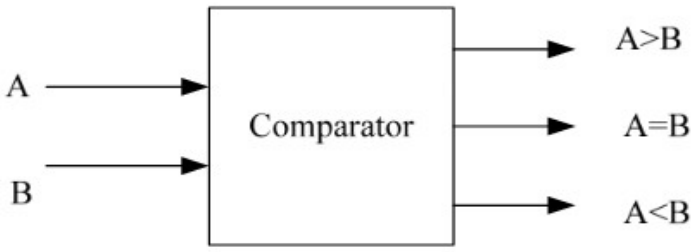
THEORY:

A **digital comparator** or **magnitude comparator** is a hardware electronic device that takes two numbers as input in binary form and determines whether one number is greater than, less than or equal to the other number. Comparators are used in Central Processing Units (CPUs) and microcontroller (MCUs).



One Bit Comparator: - a magnitude *comparator* of two *1-bits*, (A and B) inputs would produce the following three output conditions when compared to each other.

Logic Diagram:



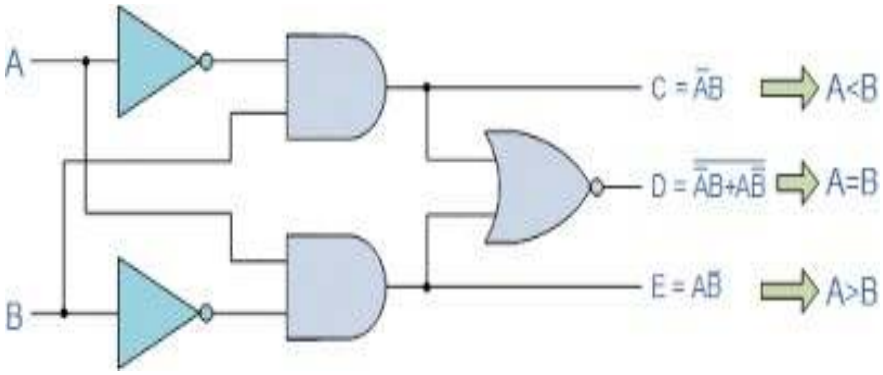
Truth Table:

A	**B**	**A>B**	**A<B**	**A=B**
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

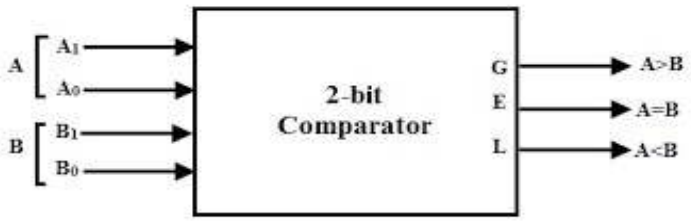
A>B = AB’

A<B = A'B

(A=B) = A’B’ + AB = A EXOR B



Two Bit Comparator: - a magnitude *comparator* of two 2-*bits*, (A and B) inputs would produce the following three output conditions when compared to each other.

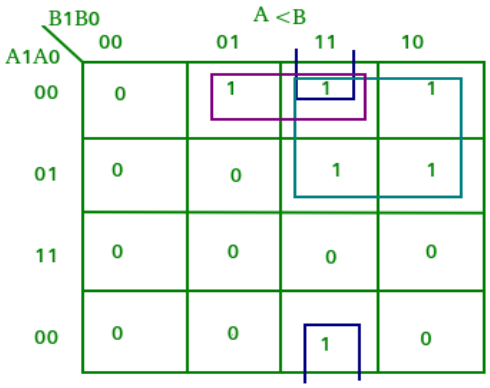


Truth Table:

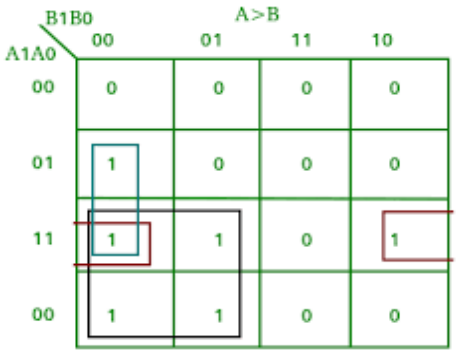
A1	A0	B1	B0	A>B	A<B	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Kmap:

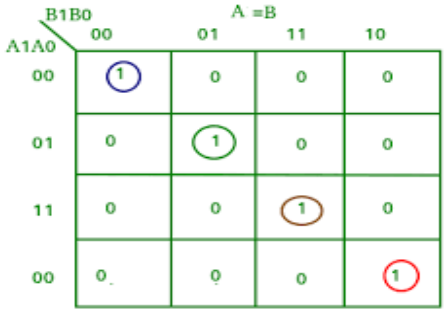
Kmap for A<B



Kmap for A>B



Kmap A=B



$$A>B:\ G = A0\,\overline{B1}\ \ \overline{B0} + A1\,\overline{B1} + A1\ A0\,\overline{B0}$$

$$A = B:\ E = \overline{A1}\ \ \overline{A0}\ \overline{B1}\ \ \overline{B0} + \overline{A1}\ A0\ \ \overline{B1}\ B0 + A1\ A0\ B1\ B0 + A1\,\overline{A0}\ B1\,\overline{B0}$$

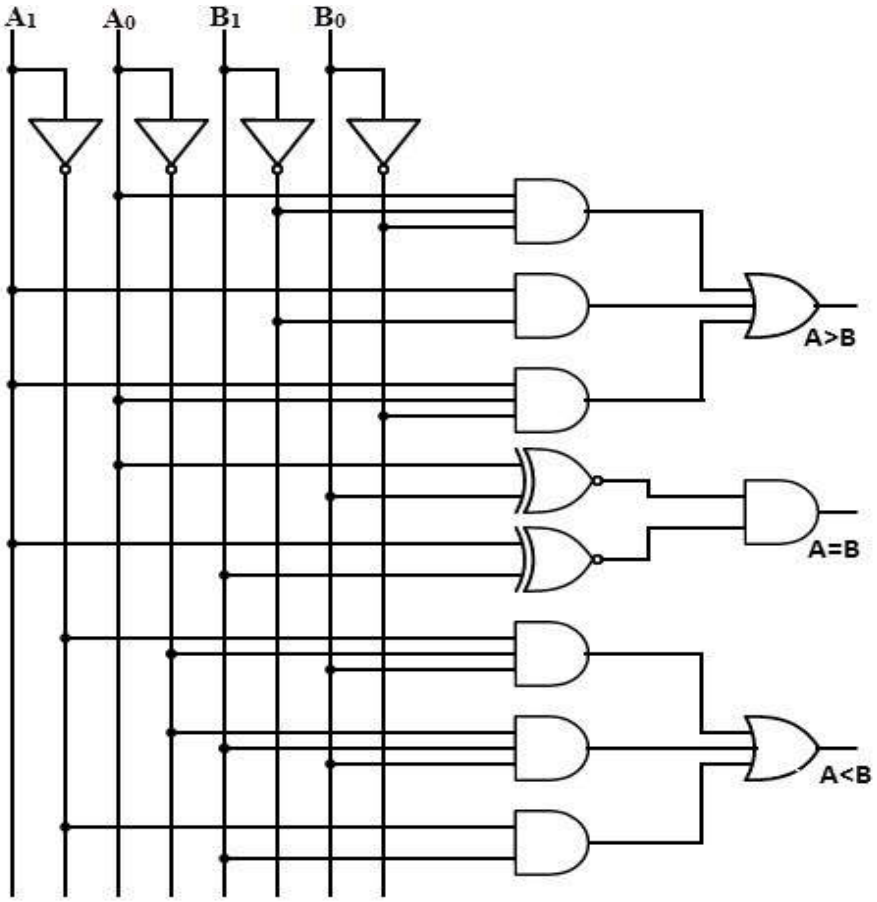
$$= \overline{A1}\ \ \overline{B1}\ (\overline{A0}\ \ \overline{B0} + A0\,B0) + A1\,B1\ (A0\,B0 + \overline{A0}\ \ \overline{B0}\ )$$

$$= (A0\,B0 + \overline{A0}\ \ \overline{B0})\ (A1\ B1 + \ \overline{A1}\ \ \overline{B1})$$

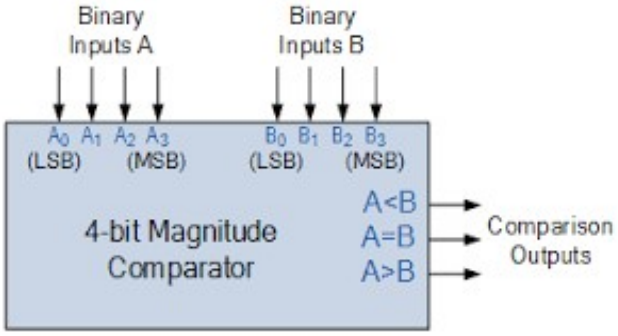
$$= (A0\ \text{Ex-NOR}\ B0)\ (A1\ \text{Ex-NOR}\ B1)$$

$$A < B:\ L = \overline{A1}\ \ B1 + \overline{A0}\ B1\ B0 + \overline{A1}\ \ \overline{A0}\ B0$$

Circuit Diagram:



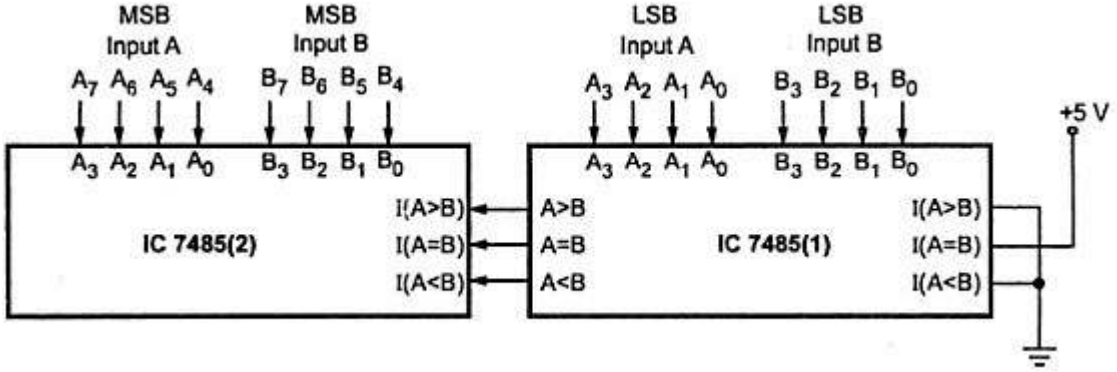
4 Bit Comparator IC 7485



The 4-bit comparator is mostly available in IC form and common type of this IC is 7485. This IC can be used to compare two 4-bit binary words by grounding I (A>B), I (A<B) and I (A=B) connector inputs to Vcc terminal. In addition to the normal comparator, this IC is provided with cascading inputs in order to facilitate the cascading several comparators. Any number of bits can be compared by cascading several of these comparator ICs.

COMPARING INPUTS				OUTPUT		
A3, B3	**A2, B2**	**A1, B1**	**A0, B0**	**A > B**	**A < B**	**A = B**
A3 > B3	X	X	X	H	L	L
A3 < B3	X	X	X	L	H	L
A3 = B3	A2 >B2	X	X	H	L	L
A3 = B3	A2 < B2	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H
H = High Voltage Level, L = Low Voltage, Level, X = Don't Care						

8-Bit Comparator using IC 7485



Applications of Comparators

1. These are used in the address decoding circuitry in computers and microprocessor based devices to select a specific input/output device for the storage of data.
2. These are used in control applications in which the binary numbers representing physical variables such as temperature, position, etc. are compared with a reference value. Then the outputs from the

comparator are used to drive the actuators so as to make the physical variables closest to the set or reference value.

3. Process controllers
4. Servo-motor control

QUESTIONS:

1. What is comparator?
2. It is combinational circuit or sequential circuit?
3. Which IC used as Magnitute comparator?

ASSIGNMENT NO: 05

TITLE: Parity Generator Checker

PROBLEM STATEMENT: Design & Implement Parity Generator using EX-OR.

APPARATUS REQUIRED:

Digital trainer kit, ICs- 7486, 74LS08, 7404 Probs

THEORY:

Parity: A term used to specify the number of one's in a digital word as odd or even. There are two types of Parity - even and odd. An even parity generator will produce a logic 1 at its output if the data word contains an odd number of ones. If the data word contains an even number of ones then the output of the parity generator will be low. By concatenating the Parity bit to the data word, a word will be formed which always has an even number of ones i.e. has even parity.

Parity bit: An extra bit attached to a binary word to make the parity of resultant word even or odd. Parity bits are extra signals which are added to a data word to enable error checking.

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	**0**0000000	**1**0000000
1010001	3	**1**1010001	**0**1010001
1101001	4	**0**1101001	**1**1101001
1111111	7	**1**1111111	**0**1111111

Parity generator: A logic circuit that generates an additional bit which when appended to a digital word makes its parity as desired (odd or even).Parity generators calculate the parity of data packets and add a parity amount to them. Parity is used on communication links (e.g. Modem lines) and is often included in memory systems. If a data word is sent out with even parity, but has odd parity when it is received then the data has been corrupted and must be resent. As its name implies the operation of an Odd Parity generator is similar but it provides odd parity. The table shows the parity generator outputs for various 8-bit data words.

Parity checker: A logic circuit that checks the parity of binary word. Parity checkers are integrated circuits (ICs) used in digital systems to detect errors when streams of bits are sent from a transmitter to a receiver. The minimum distance between any two code words with parity bit attached is two. The parity bit 1 or 0 is attached to the code to be transmitted at the transmitter end and the parity of the received (n+1)-bit word is checked at the receiving end. If there is only one error, the erroneously code is detected at the receiving end by parity check.

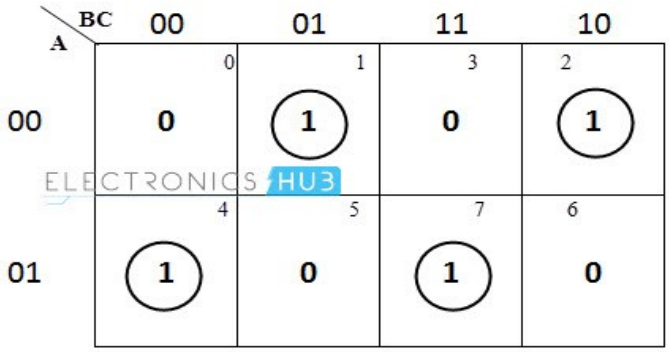
Design:

3 Bit Even Parity Generator

the truth table of even parity generator in which 1 is placed as parity bit in order to make all 1s as even when the number of 1s in the truth table is odd.

3-bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The K-map simplification for 3-bit message even parity generator is



From the above truth table, the simplified expression of the parity bit can be written as

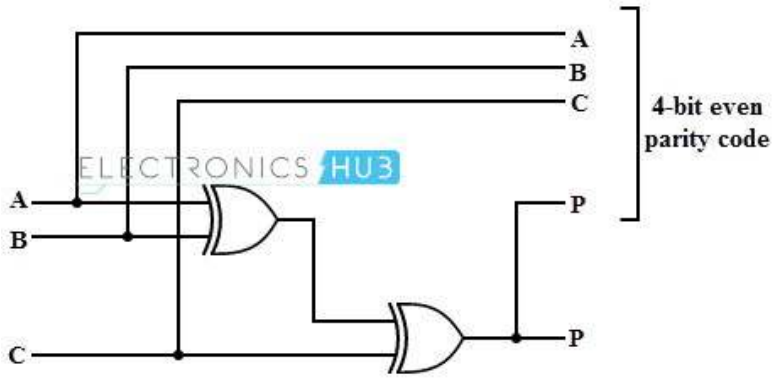
$$P = \bar{A}\,\bar{B}\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,C$$

$$= \bar{A}\,(\bar{B}\,C + B\,\bar{C}) + A\,(\bar{B}\,\bar{C} + B\,C)$$

$$= \bar{A}\,(B \oplus C) + A\,(\overline{B \oplus C})$$

$$P = A \oplus B \oplus C$$

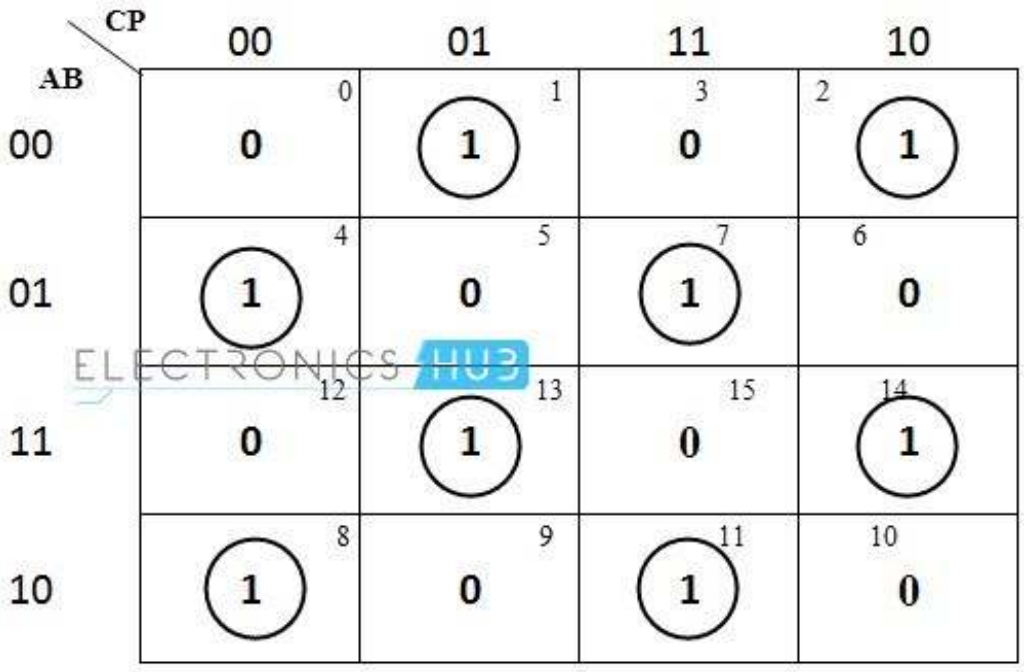
Digital Circuit:



Even Parity Checker

4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

The above truth table can be simplified using K-map as shown below.



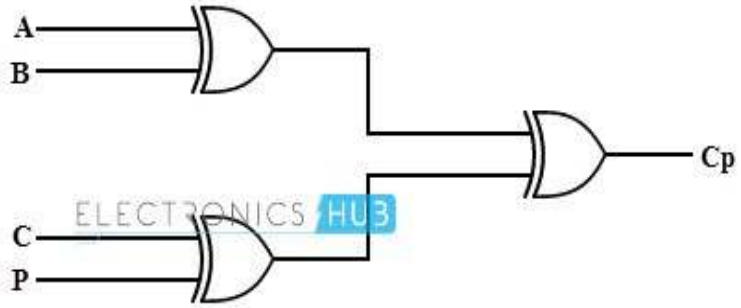
$$PEC = \bar{A}\ \bar{B}\ (\bar{C}\,D + C\,\bar{D}) + \bar{A}\,B\ (\bar{C}\,\bar{D} + C\,D) + A\,B\ (\bar{C}\,D + C\,\bar{D}) + A\,\bar{B}\ (\bar{C}\,\bar{D} + C\,D)$$

$$= \bar{A}\ \bar{B}\ (C \oplus D) + \bar{A}\,B\ (\overline{C \oplus D}) + A\,B\ (C \oplus D) + A\,\bar{B}\ (\overline{C \oplus D})$$

$$= (\bar{A}\ \bar{B} + A\,B)\ (C \oplus D) + (\bar{A}\,B + A\,\bar{B})\ (\overline{C \oplus D})$$

$$= (A \oplus B) \oplus (C \oplus D)$$

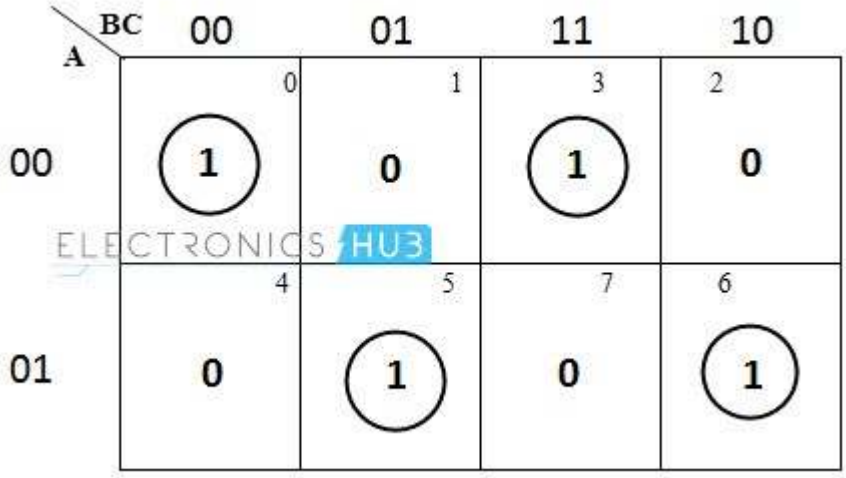
The above logic expression for the even parity checker can be implemented by using three Ex-OR gates as shown in figure. If the received message consists of five bits, then one more Ex-OR gate is required for the even parity checking.



Odd Parity Generator

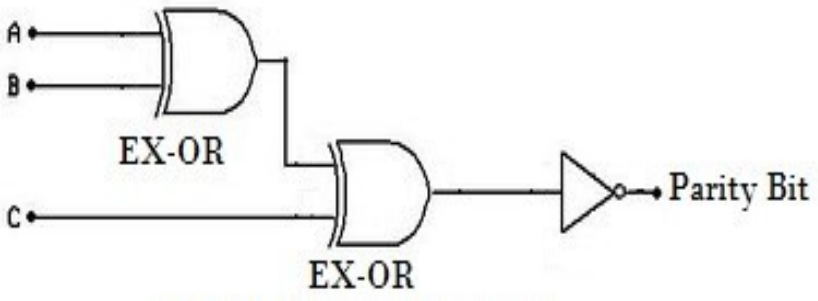
3-bit message			Odd parity bit generator (P)
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The truth table of the odd parity generator can be simplified by using K-map as



The output parity bit expression for this generator circuit is obtained as

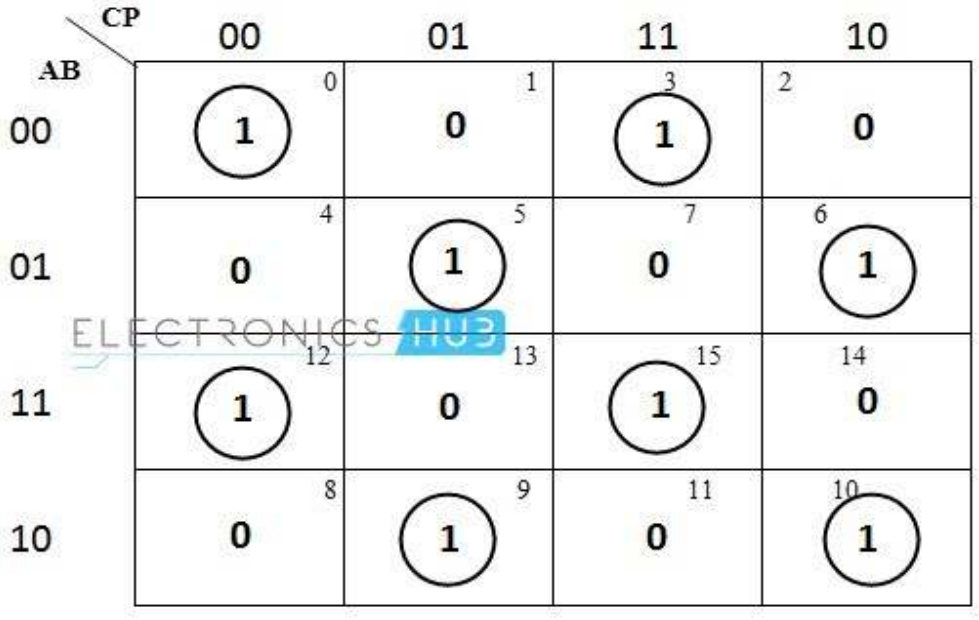
$$P = A \oplus B \text{ Ex-NOR } C$$



Odd Parity Checker

4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

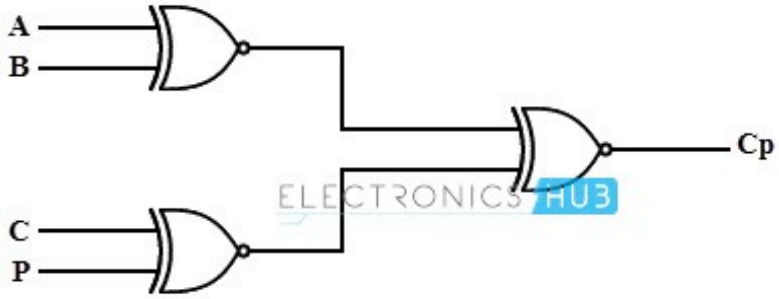
The expression for the PEC in the above truth table can be simplified by K-map as shown below.



After simplification, the final expression for the PEC is obtained as

PEC = (A Ex-NOR B) Ex-NOR (C Ex-NOR D)

The expression for the odd parity checker can be designed by using three Ex-NOR gates as shown below.

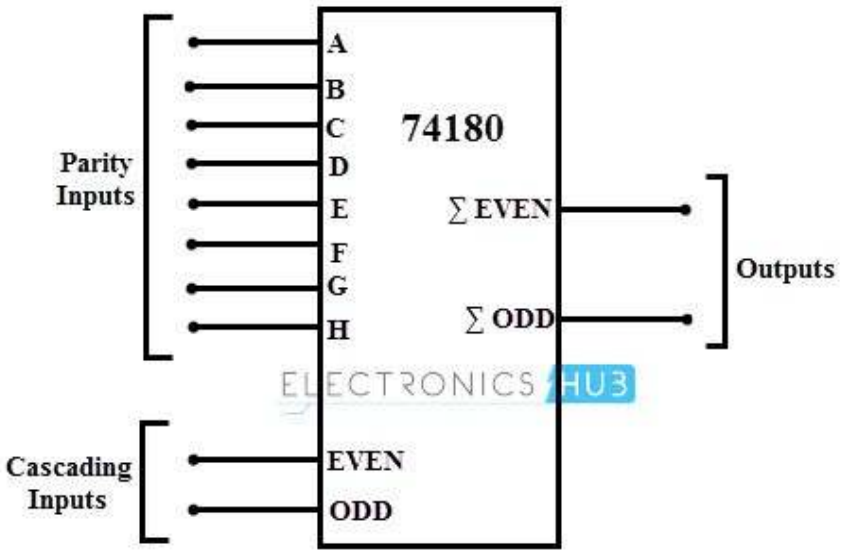


Parity Generator/Checker ICs

There are different types of parity generator /checker ICs are available with different input configurations such as 5-bit, 4-bit, 9-bit, 12-bit, etc. A most commonly used and standard type of parity generator/checker IC is 74180.

It is a 9-bit parity generator or checker used to detect errors in high speed data transmission or data retrieval systems. The figure below shows the pin diagram of 74180 IC.

This IC can be used to generate a 9-bit odd or even parity code or it can be used to check for odd or even parity in a 9-bit code (8 data bits and one parity bit).



This IC consists of eight parity inputs from A through H and two cascading inputs. There are two outputs even sum and odd sum. In implementing generator or checker circuits, unused parity bits must be tied to logic zero and the cascading inputs must not be equal.

Questions:

1. What is parity bit?
2. What is even parity and odd parity?
3. What is the use of parity bit?
4. What are the other error detecting codes and correcting?
5. What is IC 74180?
6. What is the disadvantage of parity bit used for error detection?

ASSIGNMENT NO: 06

TITLE: Flip flop Conversion

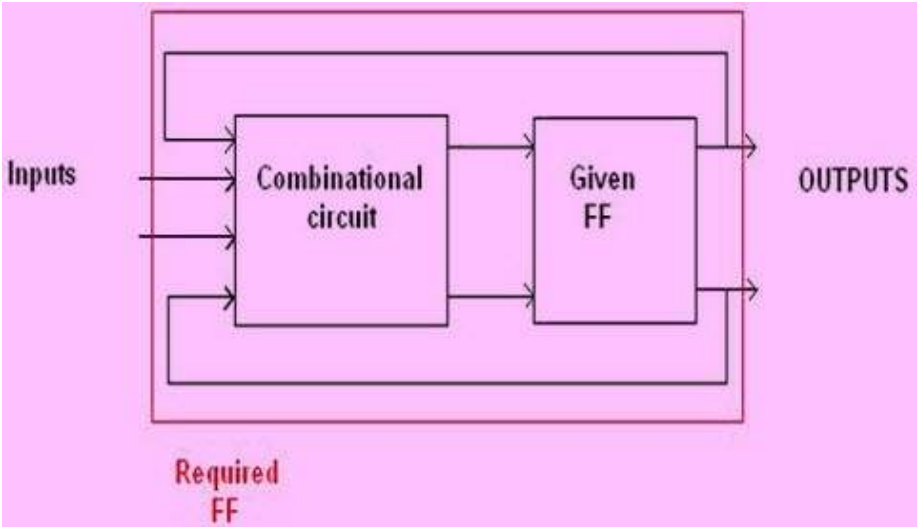
PROBLEM STATEMENT: Design & realization of sequential circuits for flip flop conversion

PRE-REQUISITE:

Digital trainer kit, ICs- 7474, 7476, 74LS08, 7404 , 7432, Probs

THEORY:

In electronics, flip flop is an electronic circuit and is is also called as a latch. Flip flops consist of two stable states which are used to store the data. These are basic building blocks of a digital electronics system which are used in various systems like communications, computers, etc. A basic flip flop can be used to construct a cross coupled inverting elements like invert gates, FETs, BJTs, inverters, vacuum tubes. Conversion of one type of flip flop to another can be done by using a combinational logic circuit. For instance, If a JK Flip Flop is necessary, the i/ps are given to the combinational circuit & the o/p of the combinational circuit is given to the i/ps of the actual flip-flop. Therefore, the o/p of the actual flip-flop is the o/p of the required flip-flop.



The main purpose of the flip flop conversion is to convert a flip flop into a desired type-B flip flop using some conversion logic. The flip flop conversions are classified into different types that are as follows,

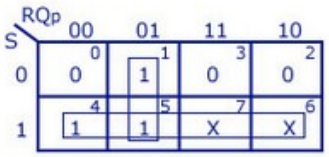
1. D-FF to SR-FF Conversion

D is the actual input of the flip flop and S and R are the external inputs. Eight possible combinations are achieved from the external inputs S, R and Qp. But, since the combination of S=1 and R=1 are invalid, the values of Qp+1 and D are considered as “don’t cares”. The logic diagram showing the conversion from D to SR, and the K-map for D in terms of S, R and Qp are shown below.

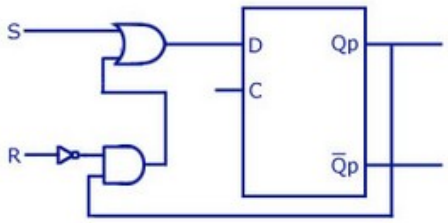
Conversion Table

S-R Inputs		Outputs		D Input
S	R	Qp	Qp+1	
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	Invalid		Dont care
1	1	Invalid		Dont care

K-map



Logic Diagram



2. **D Flip Flop to JK Flip Flop**

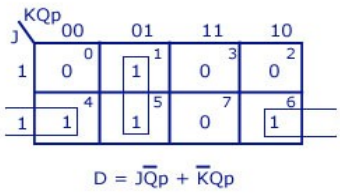
In this conversion, D is the actual input to the flip flop and J and K are the external inputs. J, K and Qp make eight possible combinations, as shown in the conversion table below. D is expressed in terms of J, K and Qp. The conversion table, the K-map for D in terms of J, K and Qp and the logic diagram showing the conversion from D to JK are given in the figure below.

D Flip Flop to J-K Flip Flop

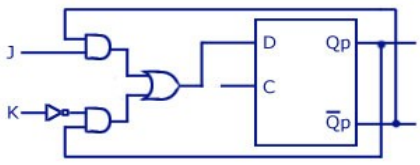
Conversion Table

J-K Input		Outputs		D Input
J	K	Qp	Qp+1	
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

K-map



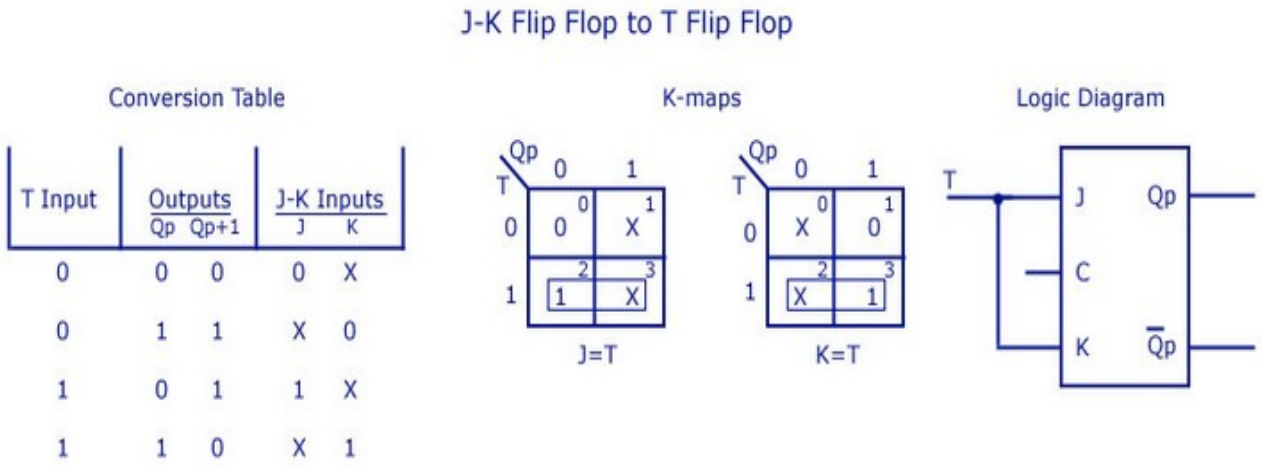
Logic Diagram



www.CircuitsToday.com

3. **JK Flip Flop to T Flip Flop**

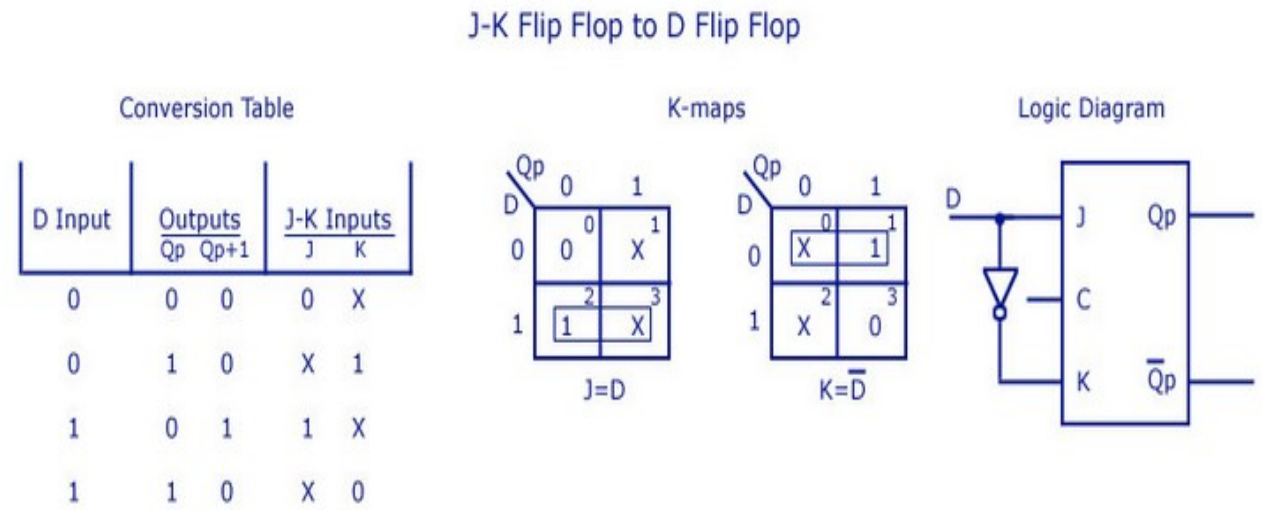
J and K are the actual inputs of the flip flop and T is taken as the external input for conversion. Four combinations are produced with T and Qp. J and K are expressed in terms of T and Qp. The conversion table, K-maps, and the logic diagram are given below.



T Input	Outputs Qp	Outputs Qp+1	J-K Inputs J	J-K Inputs K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

4. **JK Flip Flop to D Flip Flop**

D is the external input and J and K are the actual inputs of the flip flop. D and Qp make four combinations. J and K are expressed in terms of D and Qp. The four combination conversion table, the K-maps for J and K in terms of D and Qp, and the logic diagram showing the conversion from JK to D are given below.



D Input	Outputs Qp	Outputs Qp+1	J-K Inputs J	J-K Inputs K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	0	X	0

QUESTIONS:

1. What is Flip-Flop?
2. FF is sequesntial ckt or combinational ckt?
3. What are different types of FF?
4. What is triggering? Explain Methos.
5. Why T FF called Toggle FF?
6. What is RACE condition in S-R FF?
7. Applications of FF.

ASSIGNMENT NO: 07

TITLE: Ripple Counter

PROBLEM STATEMENT: Design of Ripple Counter using suitable Flip Flops

PRE-REQUISITE:

Digital trainer kit, ICs- 7474, 7476 Probs

THEORY:

A counter is basically used to count the number of clock pulses applied to a flip-flop. It can also be used for Frequency divider, time measurement, frequency measurement, distance measurement and also for generating square waveforms. In this, the flip-flops are asynchronous counters and are supplied with different clock signals, there may be a delay in producing output. Also, a few numbers of logic gates are needed to design asynchronous counters. So they are elementary in design and also are less expensive.

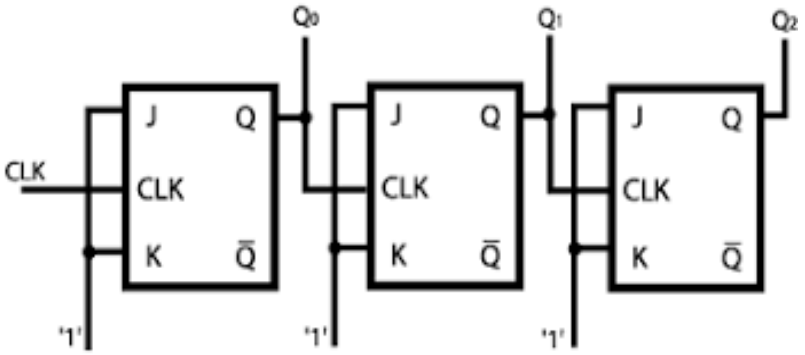
Ripple counter –

A n-bit ripple counter can count up to 2^n states. It is also known as MOD n counter. It is known as ripple counter because of the way the clock pulse ripples its way through the flip-flops. Some of the features of ripple counter are:

1. It is an asynchronous counter.
2. Different flip-flops are used with a different clock pulse.
3. All the flip-flops are used in toggle mode.
4. Only one flip-flop is applied with an external clock pulse and another flip-flop clock is obtained from the output of the previous flip-flop.
5. The flip-flop applied with external clock pulse act as LSB (Least Significant Bit) in the counting sequence.

A counter may be an up counter that counts upwards or can be a down counter that counts downwards or can do both i.e.count up as well as count downwards depending on the input control. The sequence of counting usually gets repeated after a limit. When counting up, for n-bit counter the count sequence goes from 000, 001, 010, … 110, 111, 000, 001, … etc. When counting down the count sequence goes in the opposite manner: 111, 110, … 010, 001, 000, 111, 110, … etc.

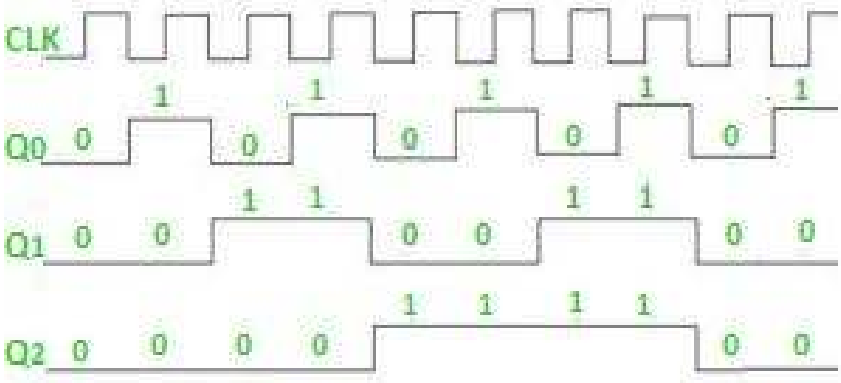
A 3-bit UP Ripple counter using JK flip-flop



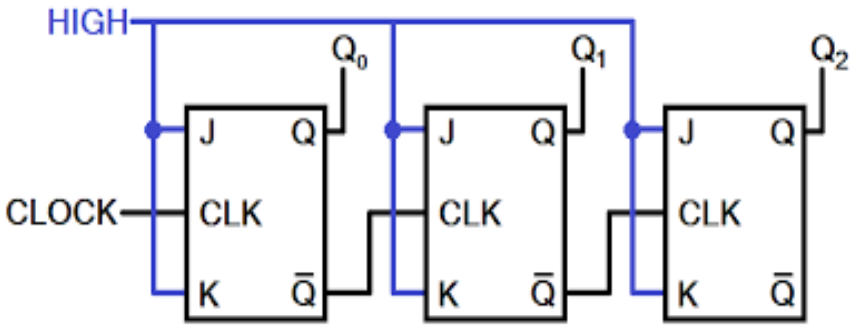
Truth Table:

Counter State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Timing diagram – Let us assume that the clock is negative edge triggered so above counter will act as an up counter because the clock is negative edge triggered and output is taken from Q.



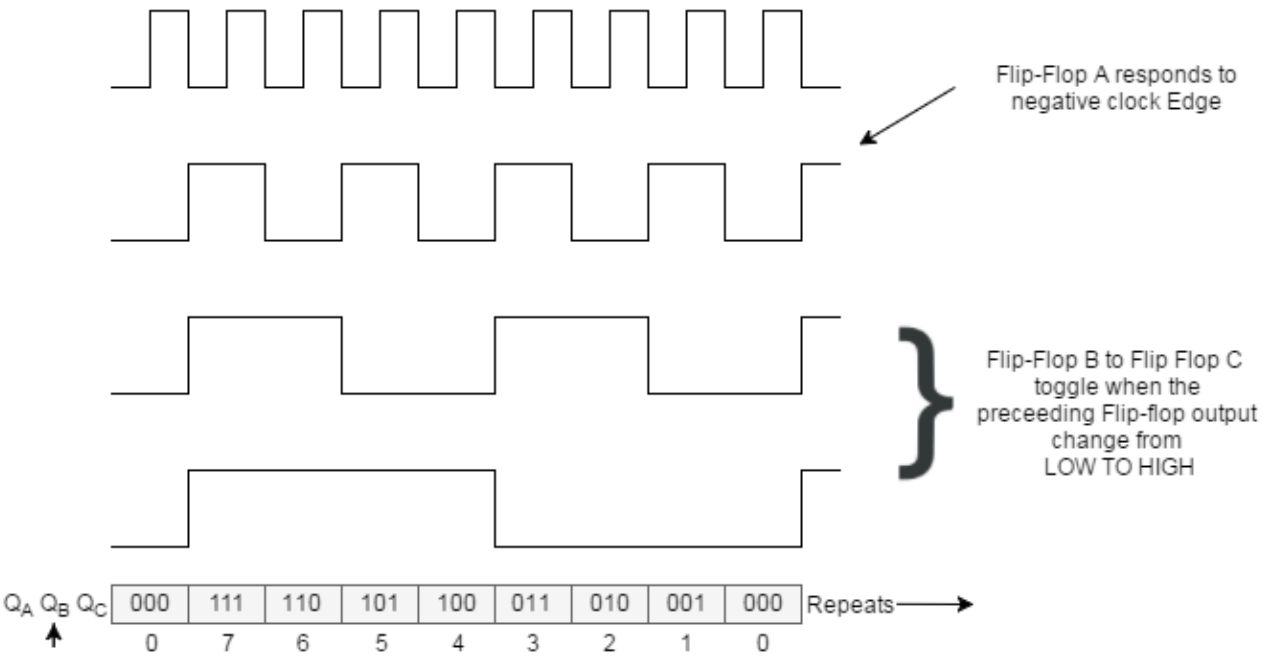
A 3-bit DOWN Ripple counter using JK flip-flop



Truth Table:

Counter state	Q_2	Q_1	Q_0
0	0	0	0
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

Timing Diagram:



Timing Diagram of a 3bit Down Counter

QUESTIONS:

1. What is counter?
2. Types of counter?
3. What ais asynchronous counter?
4. Differnce between UP and DOWN counter ckt.

ASSIGNMENT NO: 08

TITLE: Synchronous Counter & Modulo N counter

PROBLEM STATEMENT: a. Realization of 3 bit Up/Down Counter using MS JK Flip Flop / D FF

b. Realization of Mod -N counter using 7490

PRE-REQUISITE:

Digital Trainer Kit, ,IC 7474,IC 7476,IC 7408 ,IC 7490 (Decade Counter IC), Patch Cord, + 5V Power Supply

THEORY:

Counter: counters are logical device or registers capable of counting the no of states or no of clock pulse arriving at its clock input where clock is a timing parameter arriving at regular intervals of time, so counters can be also used to measure time & frequencies. They are made up of flip flops. Where the pulse are counted to be made of it goes up step by step & the o/p of counter in the flip flop is decoded to read the count to its starting step after counting n pulse incase of module & counters.

Synchronous Counter: Synchronous Counters are so called because the clock input of all the individual flip-flops within the counter are all clocked together at the same time by the same clock signal. With the Synchronous Counter, the external clock signal is connected to the clock input of EVERY individual flip-flop within the counter so that all of the flip-flops are clocked together simultaneously (in parallel) at the same time giving a fixed time relationship. In other words, changes in the output occur in "synchronisation" with the clock signal. The result of this synchronisation is that all the individual output bits changing state at exactly the same time in response to the common clock signal with no ripple effect and therefore, no propagation delay.

Ex:- Ring counter & Johnson counter

Bidirectional Counter

Both Synchronous and Asynchronous counters are capable of counting "Up" or counting "Down", but their is another more "Universal" type of counter that can count in both directions either Up or Down depending on the state of their input control pin and these are known as Bidirectional Counters. Bidirectional counters, also known as Up/Down counters, are capable of counting in either direction through any given count sequence and they can be reversed at any point within their count sequence by using an additional control input as shown below.

Synchronous 3-bit Up/Down Counter

Excitation Table :- The tabular representation of the operation of flip flop (i.e: Operational Characteristic)

Present state	Next state	J	K
0	1	0	X
0	1	1	X
1	0	X	1
1	1	X	0

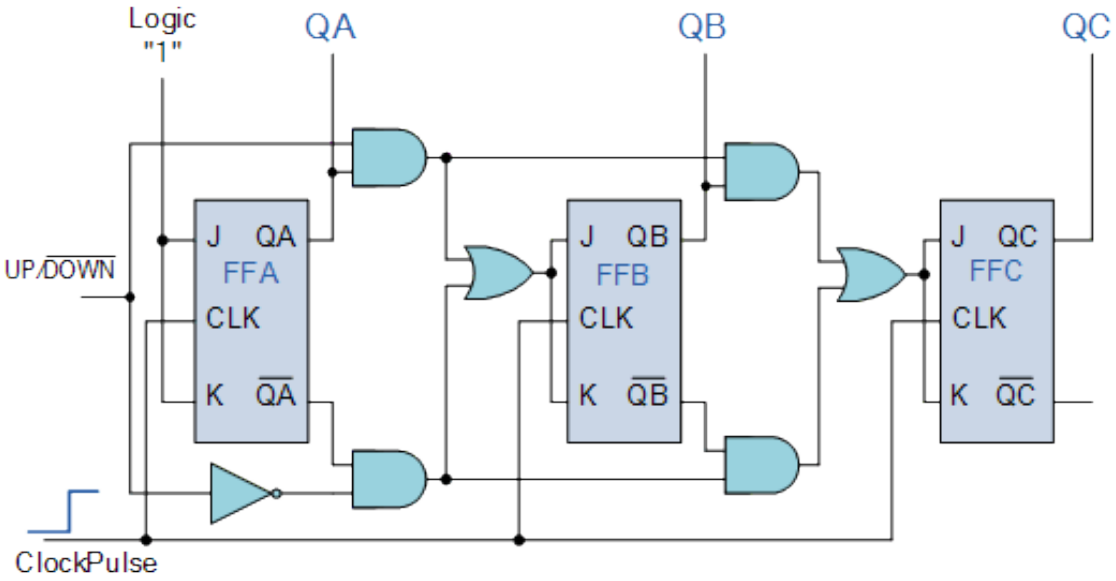
For M = 0, it acts as an Up counter and for M =1 as an Down counter. State Table for 3 bit Up-Down Synchronous Counter :

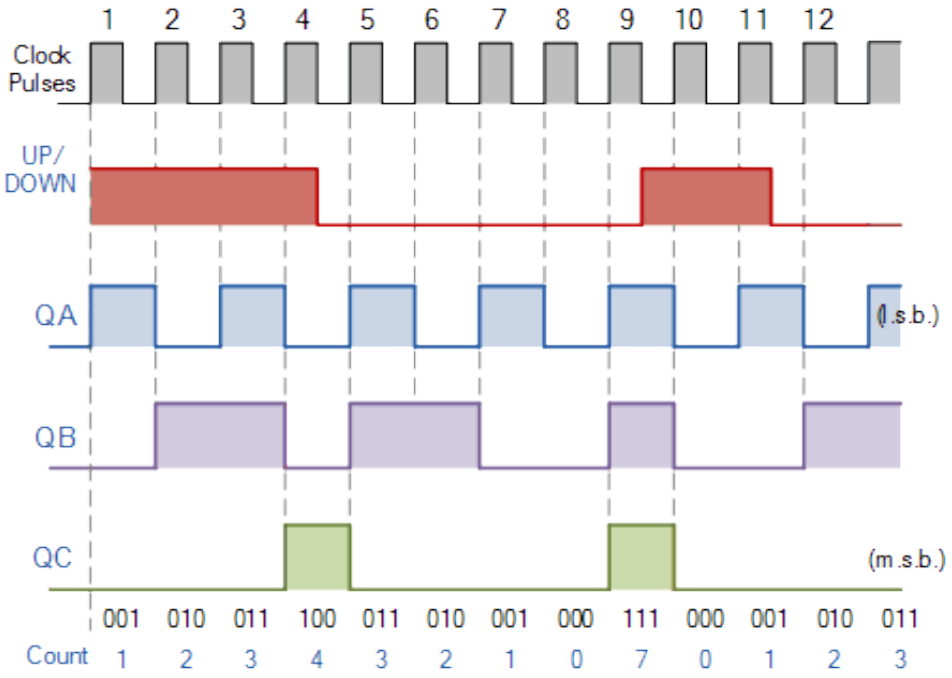
Control input M	Present State			Next State			Input for Flip-flop					
	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	1	0	X	0	X	1	X
0	0	0	1	0	1	0	0	X	1	X	X	1
0	0	1	0	0	1	1	0	X	X	0	1	X
0	0	1	1	1	0	0	1	X	X	1	X	1
0	1	0	0	1	0	1	X	0	0	X	1	X
0	1	0	1	1	1	0	X	0	1	X	X	1
0	1	1	0	1	1	1	X	0	X	0	1	X
0	1	1	1	0	0	0	X	1	X	1	X	1
1	0	0	0	1	1	1	1	X	1	X	X	1
1	0	0	1	0	0	0	0	X	0	X	1	X
1	0	1	0	0	0	1	0	X	X	1	X	1
1	0	1	1	0	1	0	0	X	X	0	1	X
1	1	0	0	0	1	1	X	1	1	X	X	1
1	1	0	1	1	0	0	X	0	0	X	1	X
1	1	1	0	1	0	1	X	0	X	1	X	1
1	1	1	1	1	1	0	X	0	X	0	1	X

$J_A = 1 \quad K_A = 1$

$J_B = M'QA + MQA' \quad K_B = M'QA + MQA'$

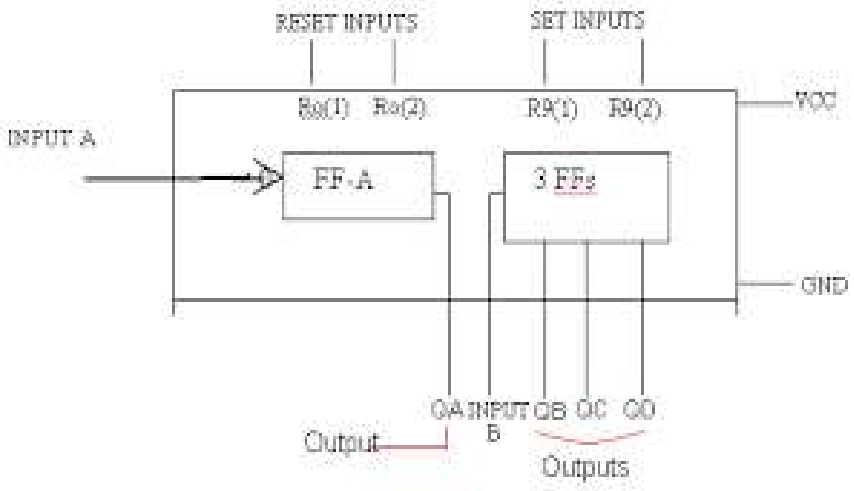
$J_C = M'QAQB + MQA'QB' \quad K_C = M'QAQB + MQA'QB'$





Mod -N counter using 7490

Ripple counter IC-7490 (decade counter): i.e .having various name like mod-n counter, decade counter, BCD counter. IC-7490 is a TTL MSI decade counter. It contains four master slave flip flops and additional gating to provide a divide-by-two counter and a three stage binary counter which provides a divide by 5 counters**.**



1. If both the reset input Ro(1) & Ro(2) are at logic 1 then all the flip-flop will be reset and the output is given by

   QD QC QB QA = 0000

2. If both the reset input R9(1) & R9(2) are at logic 1 then the counter output is set to decimal 9.

   QD QC QB QA = 1001

3. If any one pin of Ro(1) & Ro(2) and one of R9(1) & R9(2) are at low, then the counter will be in counting mode.

The reset/count function table of IC7490 as follows,

Reset inputs				**Output**			
R0(1)	**R0(2)**	**R9(0)**	**R9(1)**	**QD**	**QC**	**QB**	**QA**
1	1	0	X	0	0	0	0
1	1	X	0	0	0	0	0
X	X	1	1	1	0	0	1
X	0	X	0	COUNTER			

0	X	0	X	COUNTER
0	X	X	0	COUNTER
X	0	0	x	COUNTER

IC 7490 Decade Counter Description

Pin name	Description
Input B	This is clock input to the internal MOD-5 ripple counter, which is negative edge triggered.
R0(1),R0(2)	Gated zero reset inputs
R9(1),R9(2)	These are gated set to nine inputs
QD,QC,QB	Output of internal MOD-5 counter with QD as MSB.
QA	Output of internal MOD-2 counter with QA as LSB.
Input A	Clock input to FF-A which is negative edge triggered.

Decade Counter Operation

1. The output of MOD-2 is externally connected to the input B which is the clock input of the internal MOD-5 counter.

2. Hence QA toggles on every falling edge of clock input whereas the output QD,QC,QB of the MOD-5 counter will increment from 000 to 100 on low going change of QA output.

3. Due to cascading of MOD-2 and MOD-5 counter, the overall configuration becomes a MOD-10 i.e. decade counter.

4. The reset inputs Ro(1), Ro(2) and preset inputs R9(1), R9(2) are connected to ground so as to make them inactive.

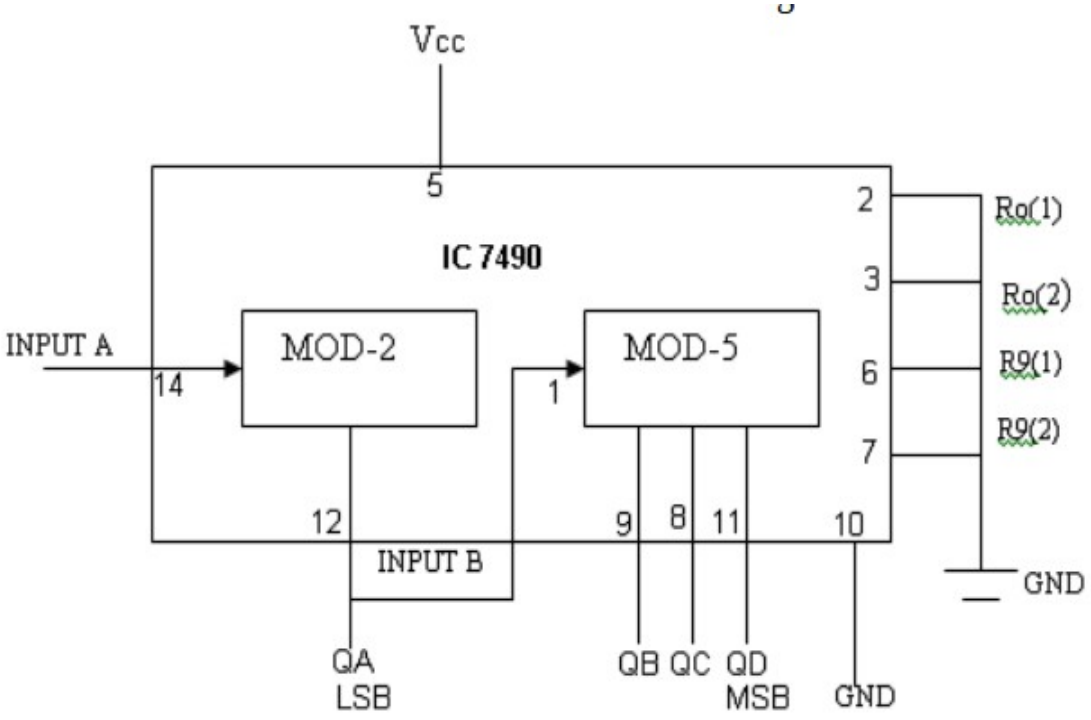
Summarizes the opération of the 7490 as décade counter

O/p of MOD-5	**O/p of MOD-2**	**CLK**	**Count**

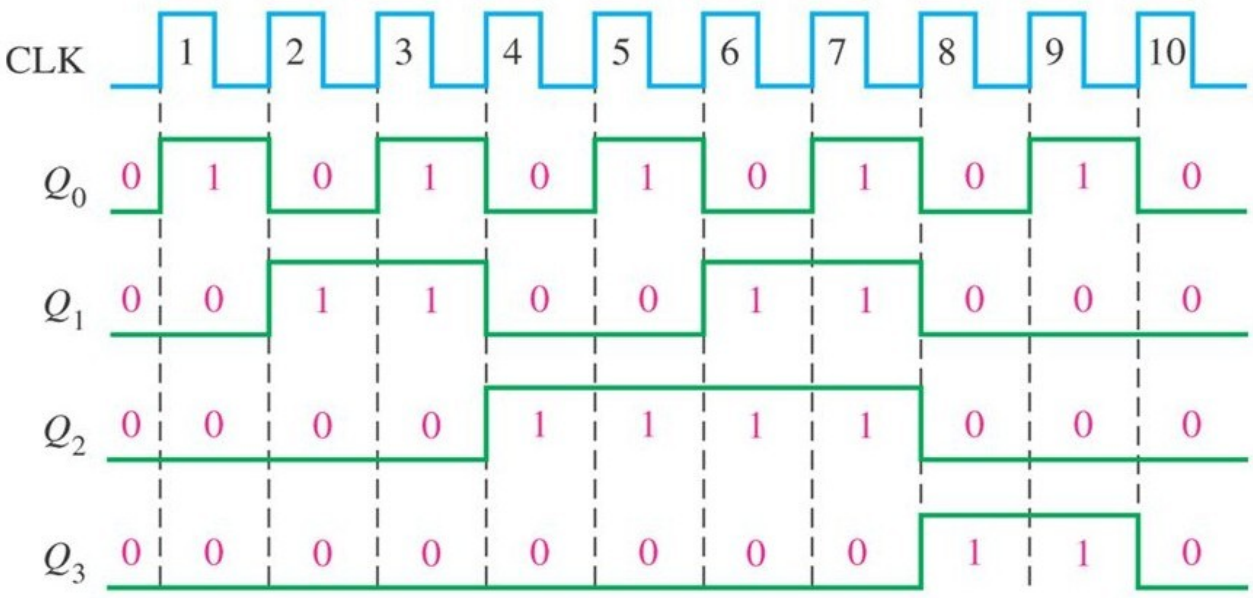
QD	QC	QB	QA		
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9

Implémentations:

Realization of MOD 10 counter using IC 7490



Timing diagram of mod10:



QUESTIONS:

1. What is synchronous counter?
2. What are advantages of synchronous counter over asynchronous counter?
3. What is Decade Counter?
4. What is Modulo N Counter?
5. What is set rest inputs in IC 7490.

ASSIGNMENT NO: 09

TITLE: Ring Counter and Jonhson Counter

PROBLEM STATEMENT: Design & Realization of Ring Counter and Johnson Ring Counter

PRE-REQUISITE:

Digital Trainer Kit, ,IC 7474,IC 7476, Patch Cord, + 5V Power Supply

THEORY:

Ring Counter

Ring counter is a typical application of Shift resister. Ring counter is almost same as the shift counter. The only change is that the output of the last flip-flop is connected to the input of the first flip-flop in case of ring counter but in case of shift resister it is taken as output. Except this all the other things are same.

No. of states in Ring counter = No. of flip-flop used

So, for designing 4-bit Ring counter we need 4 flip-flop

When PR is 0, then the output is 1. And when CLR is 0, then the output is 0. Both PR and CLR are active low signal that is always works in value 0.

PR = 0, Q = 1

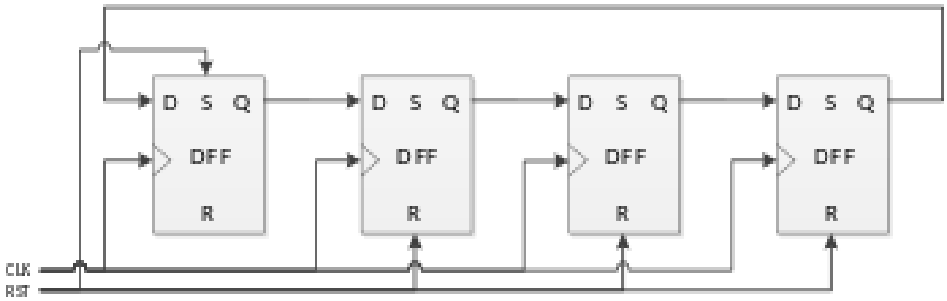
CLR = 0, Q = 0

These two values are always fixed. They are independent with the value of input D and the Clock pulse (CLK).

Working –

Here, ORI is connected to Preset (PR) in FF-0 and it is connected to Clear (CLR) in FF-1, FF-2, and FF-3. Thus, output Q = 1 is generated at FF-0 and rest of the flip-flop generate output Q = 0. This output Q = 1 at FF-0 is known as Pre-set 1 which is used to form the ring in the Ring Counter.

Circuit Diagram



Truth Table

PRESETED 1

ORI	CLK	Q0	Q1	Q2	Q3
low	X	1	0	0	0
high	low	0	1	0	0
high	low	0	0	1	0
high	low	0	0	0	1
high	low	1	0	0	0

Advantages

- Can be implemented using D and JK flip-flops. It is a self-decoding circuit.

Disadvantages

- Only four of the 15 states are being utilized.

Jonhson Counters

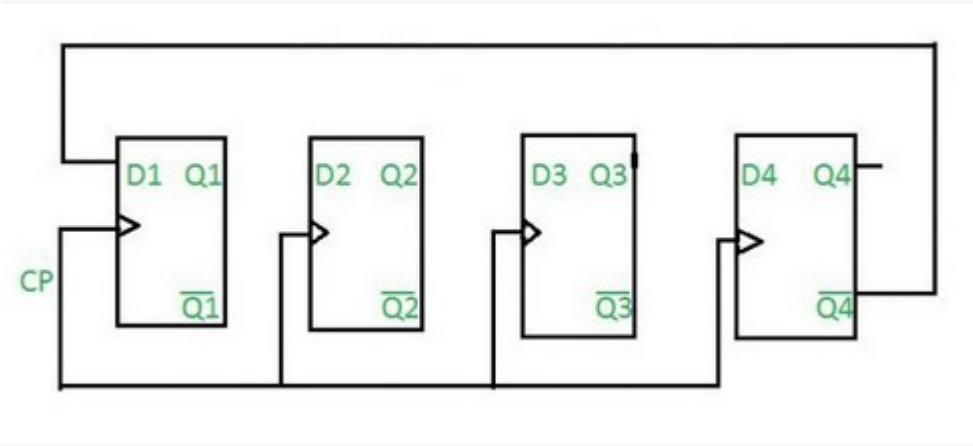
Johnson counter also known as creeping counter, is an example of synchronous counter. In Johnson counter, the complemented output of last flip flop is connected to input of first flip flop and to implement n-bit Johnson counter we require n flip-flop.It is one of the most important type of shift register counter. It is formed by the feedback of the output to its own input.Johnson counter is a ring with an inversion.Another name of Johnson counter are:creeping counter, twisted ring counter, walking counter, mobile counter and switch tail counter.

- Total number of used and unused states in n-bit Johnson counter: number of used states =2n
- number of unused states $=2^n - 2*n$

4-bit Johnson counter

Initially, suppose all flip-flops are reset.

Circuit Diagram:



Truth Table:

CP	Q1	Q2	Q3	Q4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Advantages of Johnson counter:

- The Johnson counter has same number of flip flop but it can count twice the number of states the ring counter can count.
- It can be implemented using D and JK flip flop.
- Johnson ring counter is used to count the data in a continuous loop.
- Johnson counter is a self-decoding circuit.

Disadvantages of Johnson counter:

- Johnson counter doesn't count in a binary sequence.
- In Johnson counter more number of states remain unutilized than the number of states being utilized.
- The number of flip flops needed is one half the number of timing signals.
- It can be constructed for any number of timing sequence.

Applications of Johnson counter:

- Johnson counter is used as a synchronous decade counter or divider circuit.
- It is used in hardware logic design to create complicated Finite states machine. ex: ASIC and FPGA design.
- The 3 stage Johnson counter is used as a 3 phase square wave generator which produces 1200 phase shift.
- It is used to divide the frequency of the clock signal by varrying their feedback.

QUESTIONS:

1. What is Ring Counter?
2. What is Jonhson Counter?
3. Explain use of PRESET and CLEAR inputs of FF.
4. Ring counter is synchronous counter or asynchronous?

ASSIGNMENT NO: 10

TITLE: Sequence Generator

PROBLEM STATEMENT: Design and implement Sequence generator using JK flip- flop.

PRE-REQUISITE:

Digital Trainer Kit, ,IC 7474,IC 7476, 7408, 7432,7404,7486 Patch Cord, + 5V Power Supply

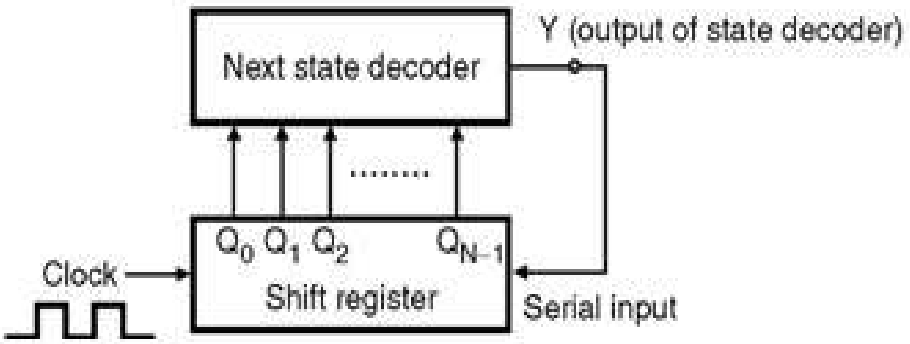
THEORY:

Sequence Generator

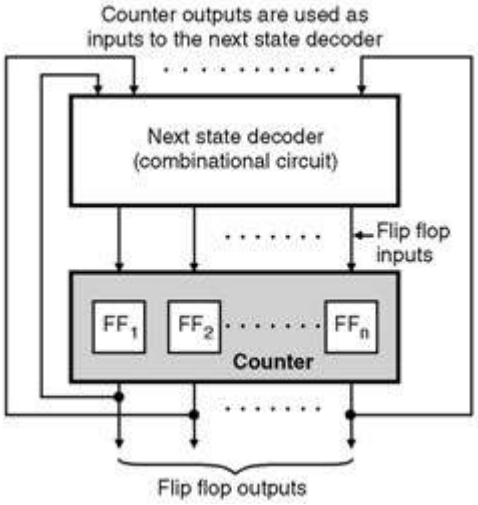
A sequence generator is a sequential circuit which generates a prescribed sequence at its output. The output sequence is in synchronication with the clock input. It is possible to design a sequence generator using counters or using the shift registers. Sequence Generator using Shift Register : The sequence generator is a circuit which generates a desired sequence bf bits at its output in synchronization with the clock. Some of the applications of the sequence generator are as follows :

1. Random bit generator
2. Counters
3. Code generators
4. Period and sequence generator.

Figure shows the basic structure of a sequence generator using Shift Register,



Sequence generator using counter is shown in Figure below,



To generate a sequence of length S it is necessary to use at least N number of

Flip-Flops, which satisfies the condition $S \leq 2^{N-1}$
The given sequence length S = 15.
Therefore N = 4.

Design a sequence generator for following sequence. Identify and check for lock out condition 0->3->5->6->0

Present State			Next State			Flip Flop inputs					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	1	1	0	x	1	X	1	x
0	0	1	x	x	x	x	x	x	X	x	x
0	1	0	x	x	x	x	x	x	X	x	x
0	1	1	1	0	1	1	x	x	1	x	0
1	0	0	x	x	x	x	x	x	X	x	x
1	0	1	1	1	0	x	0	1	X	x	1
1	1	0	0	0	0	x	1	x	1	0	x
1	1	1	x	x	x	x	x	x	X	x	x

Kmap:

i) $J_A = Q_B$

	Q_BQ_C	$Q_B\bar{Q}_C$	$\bar{Q}_B\bar{Q}_C$	$\bar{Q}_BQ_C$
Q_A	0	X	1	X
$\bar{Q}_A$	X	X	X	X

ii) $K_A = Q_B$

	Q_BQ_C	$Q_B\bar{Q}_C$	$\bar{Q}_B\bar{Q}_C$	$\bar{Q}_BQ_C$
Q_A	X	X	X	X
$\bar{Q}_A$	X	0	X	1

iii) $J_B = 1$

	Q_BQ_C	$Q_B\bar{Q}_C$	$\bar{Q}_B\bar{Q}_C$	$\bar{Q}_BQ_C$
Q_A	1	X	X	X
$\bar{Q}_A$	X	1	X	X

iv) $K_B = 1$

	Q_BQ_C	$Q_B\bar{Q}_C$	$\bar{Q}_B\bar{Q}_C$	$\bar{Q}_BQ_C$
Q_A	X	X	1	X
$\bar{Q}_A$	X	X	X	1

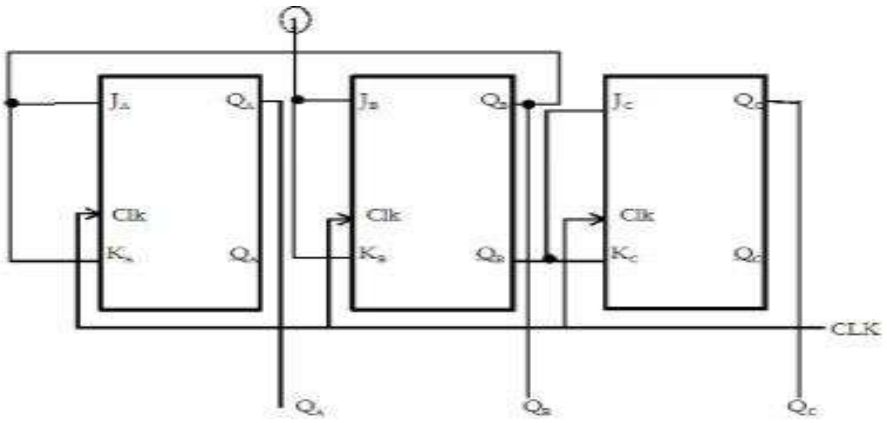
v) $J_C = Q_B$

	Q_BQ_C	$Q_B\bar{Q}_C$	$\bar{Q}_B\bar{Q}_C$	$\bar{Q}_BQ_C$
Q_A	1	X	X	X
$\bar{Q}_A$	X	X	X	0

vi) $K_C = Q_B$

	Q_BQ_C	$Q_B\bar{Q}_C$	$\bar{Q}_B\bar{Q}_C$	$\bar{Q}_BQ_C$
Q_A	X	X	0	X
$\bar{Q}_A$	X	1	X	X

Circuit Diagram:



QUESTIONS:

1. Wha are the methods used to design sequence generator?
2. What is sequence generator?
3. Sequence generator is combinational ckt or sequential ckt?
4. How to calculate no. Of FF required to design sequence generator for specific sequence?

ASSIGNMENT NO: 11

TITLE: Full Adder and MUX using VHDL

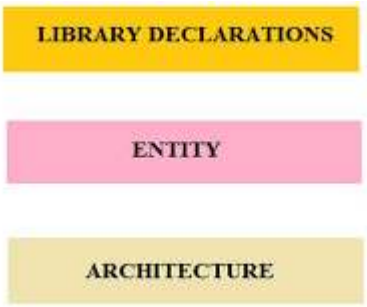
PROBLEM STATEMENT: Design and simulation of - Full adder , Flip flop, MUX using VHDL (Any 2) Use different modeling styles.

PRE-REQUISITE:

xilinx ISE

THEORY:

VHDL stands for very high-speed integrated circuit hardware description language. It is a programming language used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC program.

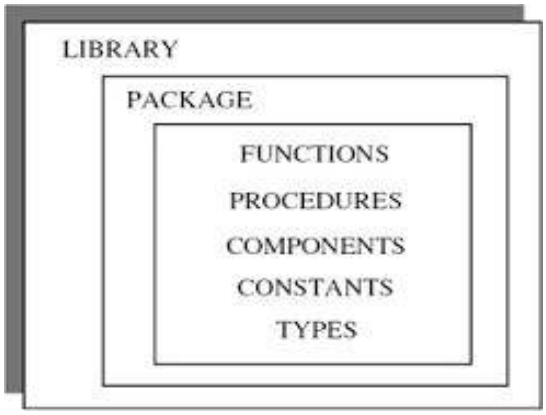


Describing a Design

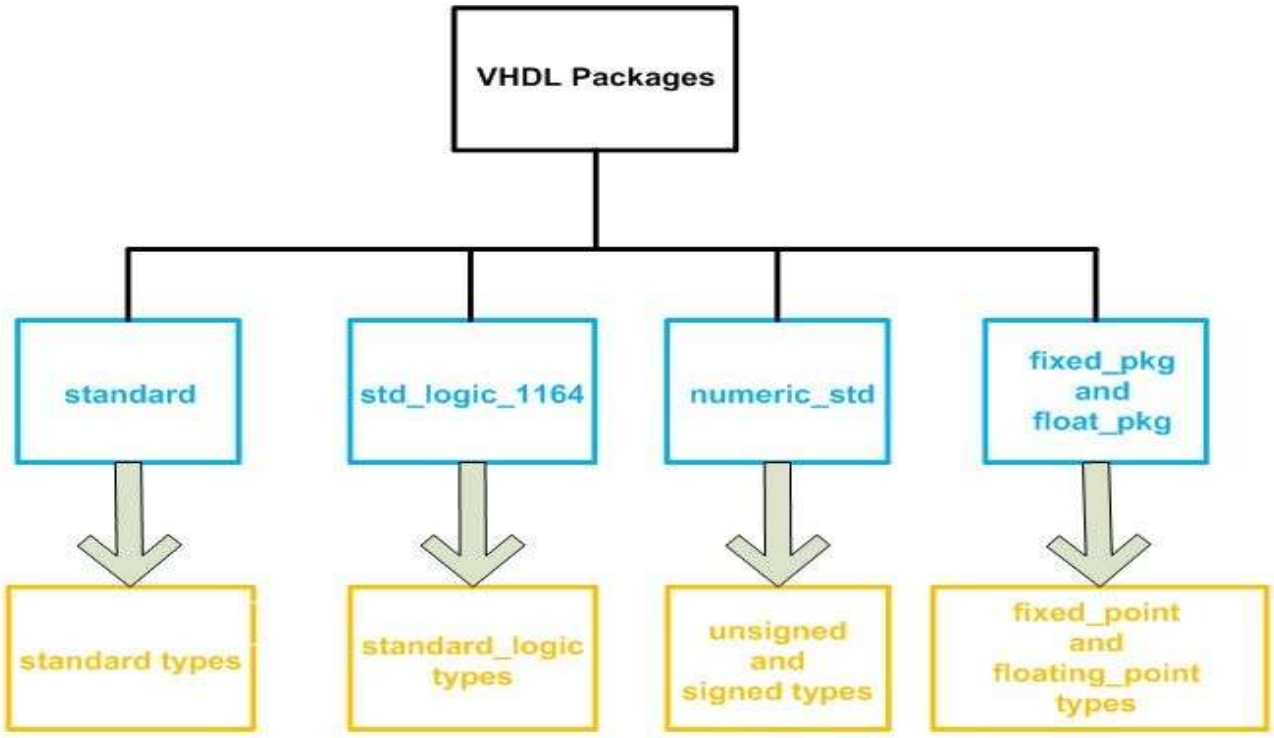
In VHDL an entity is used to describe a hardware module. An entity can be described using,

- Entity declaration
- Architecture
- Configuration
- Package declaration
- Package body

Library Declaration



Packge Description



Parts of the IEEE library can be included in an entity by inserting lines like these before your entity declaration:

library ieee;

use ieee.std_logic_1164.all;

use ieee.std_logic_arith.all;

Entity Declaration

It defines the names, input output signals and modes of a hardware module.

Syntax −

entity entity_name is

Port declaration;

end entity_name;

An entity declaration should start with ‘entity’ and end with ‘end’ keywords. The direction will be input, output or inout.

In	Port can be read
Out	Port can be written
Inout	Port can be read and written
Buffer	Port can be read and written, it can have only one source.

Architecture −

Architecture can be described using structural, dataflow, behavioral or mixed style.

Syntax −

architecture architecture_name of entity_name

architecture_declarative_part;

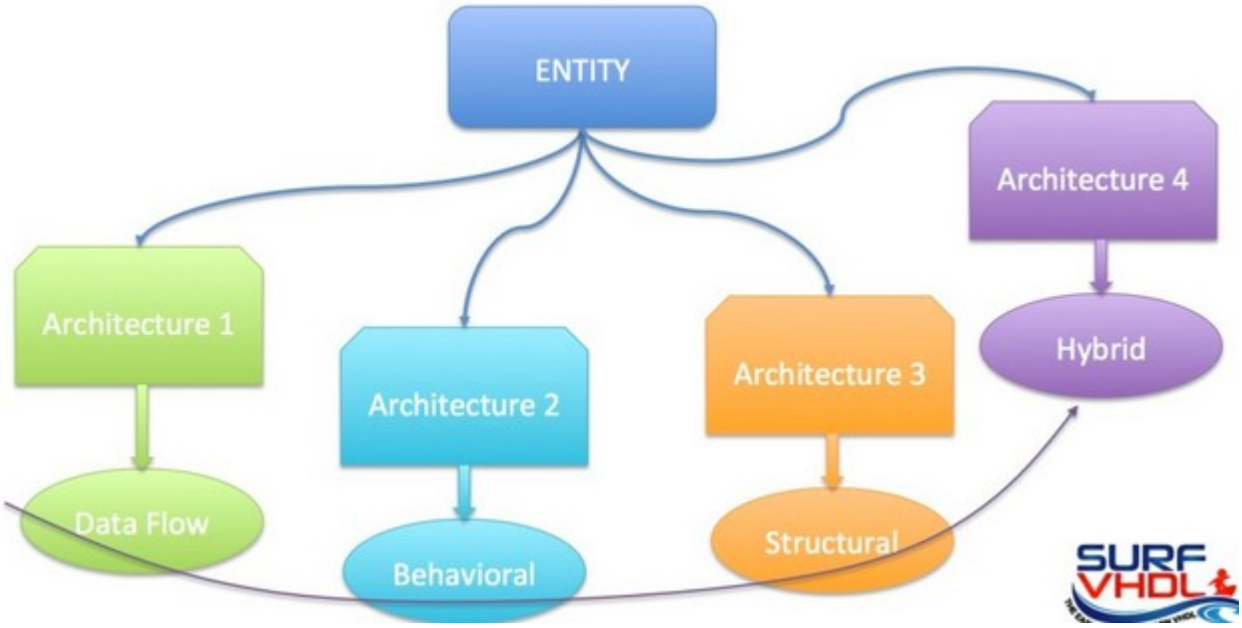
begin

Statements;

end architecture_name;

Here, we should specify the entity name for which we are writing the architecture body. The architecture statements should be inside the ‘begin’ and ‘énd’ keyword. Architecture declarative part may contain variables, constants, or component declaration.

VHDL Modeling Style



VHDL Syntax Coding Style: Behavioral, Data Flow, Structural, Hybrid

Data Flow : Concurrent Execution

Behavioural : Sequential Execution

Structural : Components Concurrent Execution

Mixed

1. Data Flow Modeling

In this modeling style, the flow of data through the entity is expressed using concurrent (parallel) signal. The concurrent statements in VHDL are WHEN and GENERATE.

Besides them, assignments using only operators (AND, NOT, +, *, sll, etc.) can also be used to construct code.

Finally, a special kind of assignment, called BLOCK, can also be employed in this kind of code.

In concurrent code, the following can be used −

- Operators
- The WHEN statement (WHEN/ELSE or WITH/SELECT/WHEN);

- The GENERATE statement;
- The BLOCK statement

2. Behavioral Modeling

In this modeling style, the behavior of an entity as set of statements is executed sequentially in the specified order. Only statements placed inside a PROCESS, FUNCTION, or PROCEDURE are sequential. PROCESSES, FUNCTIONS, and PROCEDURES are the only sections of code that are executed sequentially. However, as a whole, any of these blocks is still concurrent with any other statements placed outside it. One important aspect of behavior code is that it is not limited to sequential logic. Indeed, with it, we can build sequential circuits as well as combinational circuits.

The behavior statements are IF, WAIT, CASE, and LOOP. VARIABLES are also restricted and they are supposed to be used in sequential code only. VARIABLE can never be global, so its value cannot be passed out directly.

3. Structural Modeling

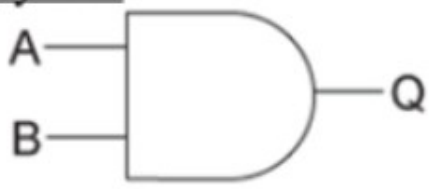
In this modeling, an entity is described as a set of interconnected components. A component instantiation statement is a concurrent statement. Therefore, the order of these statements is not important. The structural style of modeling describes only an interconnection of components (viewed as black boxes), without implying any behavior of the components themselves nor of the entity that they collectively represent.

In Structural modeling, architecture body is composed of two parts − the declarative part (before the keyword begin) and the statement part (after the keyword begin).

Logic Operation – AND GATE

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Symbol:



VHDL Code:

Library ieee;

use ieee.std_logic_1164.all;

entity and1 is

port(x,y:in bit ; z:out bit);

end and1;

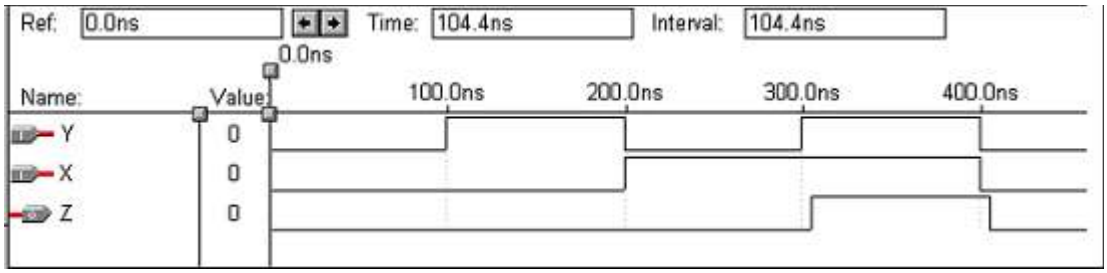
architecture virat of and1 is

begin

z<=x and y;

end virat;

Waveform:



QUESTIONS:

1. What is HDL?
2. Example of HDL?
3. Benefits of VHDL?
4. What is structure of VHDL Code?
5. What are different modelling styles?

ASSIGNMENT NO: 12

TITLE: Asynchronous counter using VHDL

PROBLEM STATEMENT: Design & simulate asynchronous counter using VHDL

PRE-REQUISITE:

xilinx ISE

THEORY:

Xilinx ISE (Integrated Synthesis Environment) is a software tool produced byXilinx for synthesis and analysis of HDL designs, enabling the developer to synthesis ("compile") their designs, perform timing analysis , examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

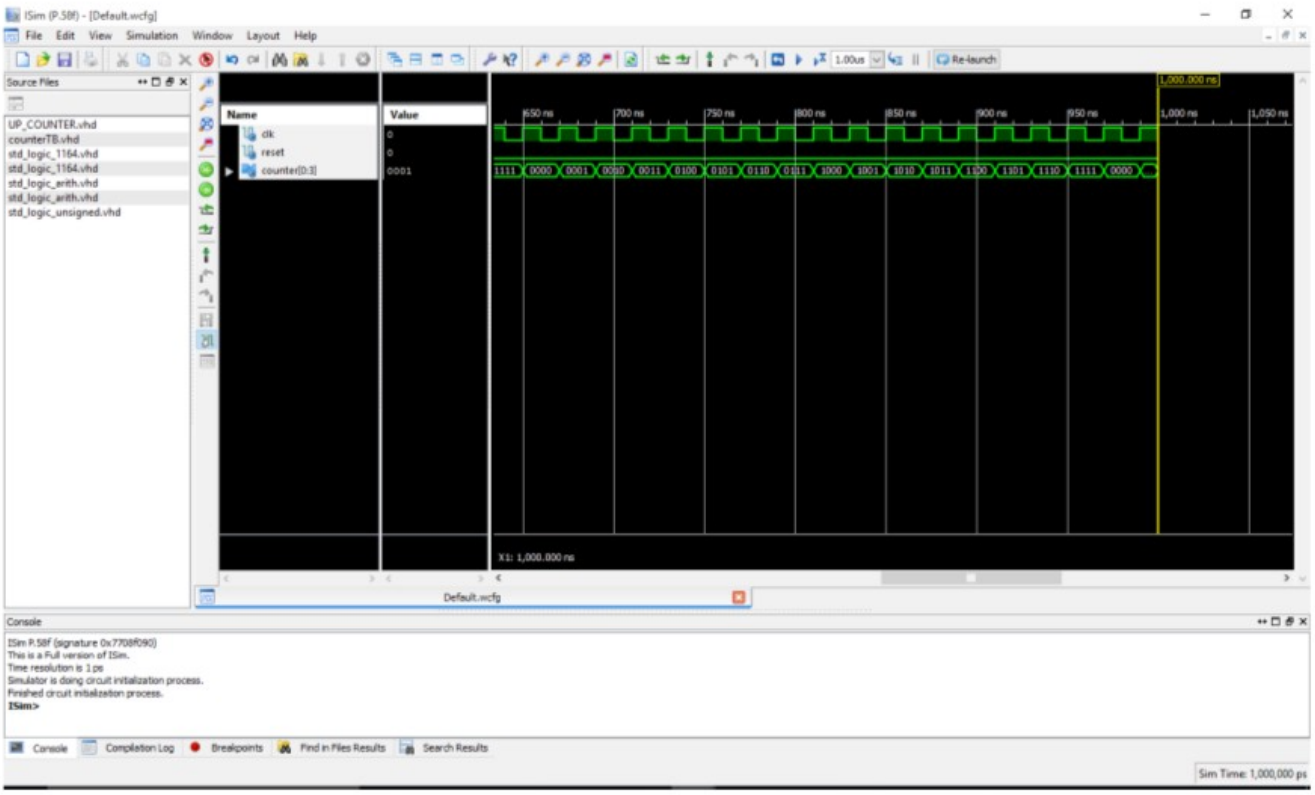
Xilinx ISE is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors.The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the ModelSim logic simulator is used for system-level testing.Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and ChipScope Pro.

Benefits of using VHDL

- Validate spec in system context (Subcontract)
- Executable specification
- Functionality separated from implementation
- Simulate early and fast (Manage complexity)
- Explore design alternatives
- Get feedback (Produce better designs)
- Automatic synthesis and test generation (ATPG for ASICs)
- Increase productivity (Shorten time-to-market)
- Technology and tool independence (though FPGA features may be unexploited)
- Portable design data (Protect investment)

Asynchronous Counter using VHDL

Waveform Screenshot



QUESTIONS:

1. Exlain the entity for AND gate.
2. What is the difference between Data Flow and Behavioral Modeling Style?
3. What are benefits of VHDL?

ASSIGNMENT NO: 13

TITLE: TTL CMOS

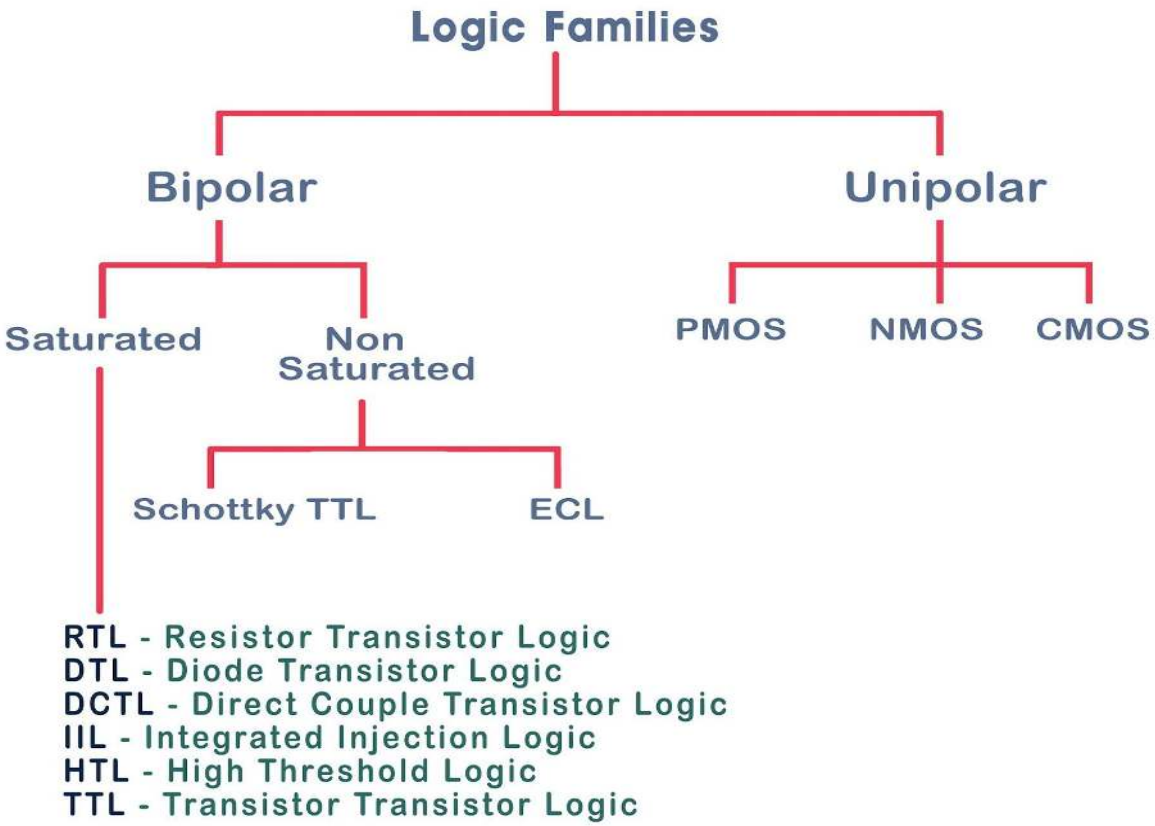
PROBLEM STATEMENT: Study of TTL Logic Family: Feature, Characteristics and Comparison with CMOS Family

PRE-REQUISITE:

Fundamental knowledge of Integrated circuits.

THEORY:

In Digital Designs, our primary aim is to create an Integrated Circuit (IC). A Circuit configuration or arrangement of the circuit elements in a special manner will result in a particular Logic Family. Electrical Characteristics of the IC will be identical. In other words, the different parameters like Noise Margin, Fan In, Fan Out etc will be identical.



Characteristics of Logic Families
The main characteristics of Logic families include:

- Speed
- Fan-in
- Fan-out
- Noise Immunity
- Power Dissipation

Speed: Speed of a logic circuit is determined by the time between the application of input and change in the output of the circuit.

Fan-in:It determines the number of inputs the logic gate can handle.

Fan-out: Determines the number of circuits that a gate can drive.

Noise Immunity: Maximum noise that a circuit can withstand without affecting the output.

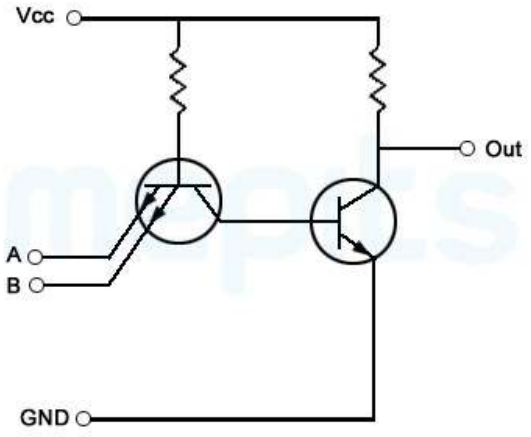
Power: When a circuit switches from one state to the other, power dissipates.

TTL Logic

In transistor-transistor logic (TTL) , logic gates and other digital circuits are made using bipolar junction transistors and resistors. The term transistor-transistor is because both logic function and amplification is done by transistor. Using TTL logic families, many logic gates can be fabricated in a single integrated circuit. For logic gate built using TTL logic families, input are given to the emitters of the input transistor. In TTL logic family, analog value from 0 V to 0.8 V is logic 0 and 2 V to 5 V is logic 1. Advantages of the TTL logic families include high switching speed (125 MHz), less noise and more current (3 mA) driving capability.

Two input NAND gate using TTL

Figure shows TTL NAND gate. There are two transistor stages in the circuit, a multi-emitter input transistor and output transistor. Function of a multi-emitter transistor is same as that of a two parallel transistor with common base and collector terminals.

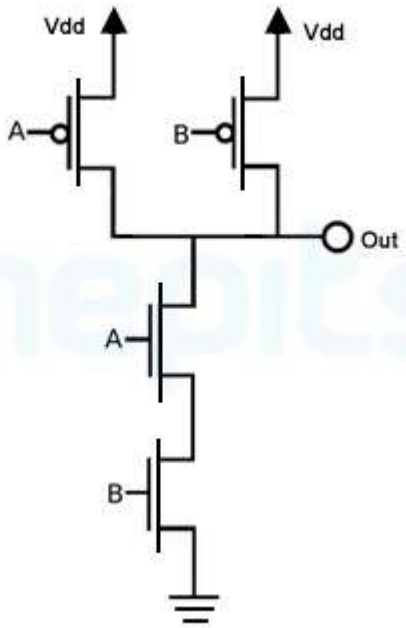


CMOS logic

Because of high noise immunity and low static power dissipation, now CMOS logic families is most preferred in large scale integrated circuits. CMOS (Complementary Metal Oxide Semiconductor) has complementary and symmetrical NMOS and PMOS transistors. Figure shown below is a **CMOS inverter.**

Depending on the input value, only one transistor of the CMOS inverter will be ON at a time. So in both states, there is no direct connection between power supply and ground, thereby reducing static power loss of a transistor.

Two input CMOS NAND gate



Difference between TTL and CMOS

The differences between TTL (transistor-transistor logic) and CMOS (Complementary Metal-oxide semiconductor) are -

1. TTL circuits use bipolar junction transistors (BJTs) while CMOS circuits use field effect transistors (FETs) ie.,by connecting NMOS and PMOS (MOSFETs).
2. A single logic gate in a CMOS chip can consist of as little as two FETs while a logic gate in a TTL chip can consist of a substantial number of parts as extra components like resistors are needed.
3. A single gate in a CMOS chip can consume around 10nW while an equivalent gate on a TTL chip can consume around 10mW of power, which is why CMOS is the preferred chip in mobile devices where power is supplied by a limited source like a battery.
4. CMOS are more susceptible to electrostatic discharge. Mere touching the terminals induce enough static electricity to damage the device permanently.
5. The basic gates used in standard TTL are NAND gates while NAND-NOR gates are used in CMOS circuits.

6. Fan-out(number of standard loads that can be connected to the output of the gate under normal operation) for TTL is 10 while it is 50 for CMOS.
7. Propagation delay for TTL is 10 ns and for CMOS, it is 70 ns.
8. Fan-in (Number of inputs that can be connected to a gate) for TTL is around 12–14 and it is greater than 10 for CMOS.
9. For TTL, the noise margin is 0.5 V while for CMOS, it is 1.5V .
10. Noise immunity of CMOS is a lot better than TTL circuits.
11. CMOS circuits are simpler to construct and has a higher packing density than TTL logic family.

QUESTIONS:

1. What is IC?
2. Classification of IS.
3. Which basic component used in TTL?
4. IC 7408 comes under which category TTL or CMOS?
5. What is fan-out and fan-in?
6. What is noise margin?

ASSIGNMENT NO: 14

TITLE: 8051 microcontroller

PROBLEM STATEMENT: Study of Microcontroller 8051 : Features, Architecture and Programming Model

PRE-REQUISITE:

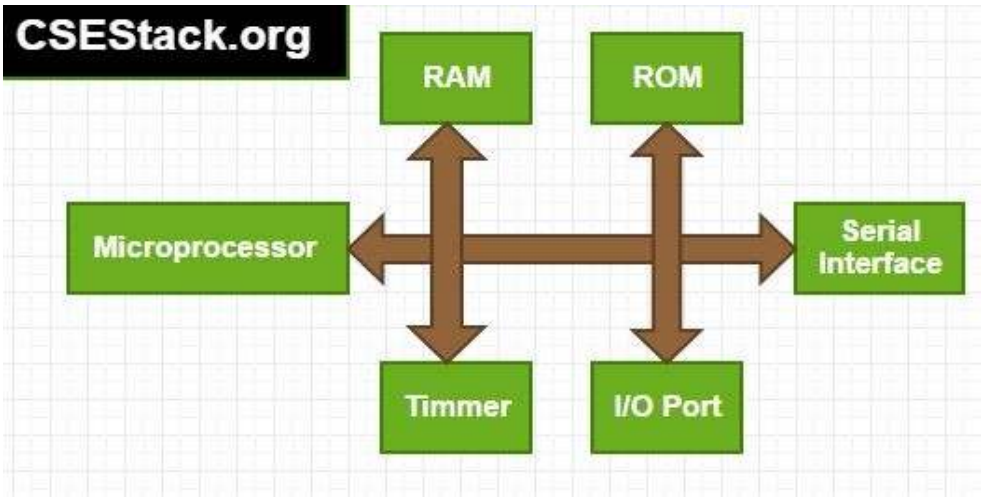
Basic knowledge of microprocessor and microcontroller.

THEORY:

Microprocessor:

The microprocessor is the electronic components. It is kind of computer processor that resides on single or multiple integrated circuits (IC). This IC contains many other electronic components such as a resistor, transistor, diode. The microprocessor performs all the functions of the central processing system. One way it differs from CPU as it occupies small space and resides on only a couple of integrated circuits. It does not have I/O peripheral component and internal memory. It requires external components such as RAM, ROM, timmer, Serial interface, I/O port to operate.

Below is the block diagram of the microprocessor architecture.



Microcontroller

The microcontroller is the electronic device. It has processing unit along with fixed sized ROM, RAM, and other required peripheral components. These all the components are embedded on the single chip.As it has all the components required to process and store data, it is also called as minicomputer or computer

on the single chip.

Below is the block diagram of the architecture of microcontroller.

Difference between Microprocessor and Microcontroller…

Computer System Vs Embedded System:

Microprocessor widely used in the computer system. And microcontroller is used in embedded system.If the microprocessor is the heart of computer system then microcontroller is the heart of the embedded system.

Architecture:

The microprocessor uses Von Neumann architecture where data and program present in the same memory module. The microcontroller uses Harvard architecture. In this module, data and program get stored in separate memory. The microcontroller can access data and program at the same time as it is in a separate memory. This is one of the reasons microcontrollers is faster than the microprocessor.

Memory and I/O Components:

The microprocessor cannot operate without peripheral components. It hs only processing unit and we have to attach all the required components externally to operate. Whereas the microcontroller has small processing unit along with internal memory to store and I/O components to give input. So it can work independently.

Circuit Size and its Complexity:

As we have to connect components externally, microprocessor circuit becomes large and complex. In a microcontroller, all the components are internally connected, so the circuit becomes too small.

Efficient Techniques to use in Compact System:

The microcontroller can be used in the compact system as it has a small size. It provides better and efficient technique in the compact system than the microprocessor.

Cost:

Microprocessor requires external components to operate. So, the Cost of the microprocessor is higher than the microcontroller.

Power Consumption:

Microprocessor requires external components and its circuits also complex. It requires more power consumption. So it is difficult to operate microprocessor using battery power.
The microcontroller has very low external components. it manages all its operation inside the single chip. So it consumes very low power supply as compared to the microprocessor. We can operate microcontroller on the externally connected stored power such as a battery.

Power Saving Feature:

The microcontroller can have multiple modes of operation such as higher performance, balance, idle or power saving mode. So if we operate microcontroller in power saving mode, the power consumption reduce even more. Most of the Microprocessor does not have this power saving feature.

Processing Speed:

The microprocessor has very less internal registers. It has to rely on external storage. So all the memory operations are carried out using memory-based external commands. Results in high processing time. The microcontroller has many registers for instruction execution. Fetching data and storing data require internal commands. So its execution and processing time are lower than the microprocessor.

Examples:

Intel Pentium series processor, core 2 duo, dual-core, Intel i3, i5 are the examples of Microprocessor that are widely used.Microcontrollers are produced by many hardware manufacturer companies such Motorola, Philips, Microchips, ATMEL…

Uses:

The microprocessor is used in desktop personal computer, laptop…
The microcontroller is used in an embedded system such as MP3 player, Television, Refrigerator, Washing Machine…

8051 Microcontroller:

The 8051 microcontroller was invented in 1980's by Intel. Its foundation is based on Harvard architecture and this microcontroller was developed principally for bringing it to be used in **Embedded Systems**. At first it was created by using NMOS technology but the use of NMOS consumed more power to work therefore Intel re-launch the microcontroller 8051 using CMOS technology and new edition came up with edition of letter 'C' in the title name, therefore the new modified version of microcontroller is called by name 80C51.

The 8051 microcontroller programming is performed in **embedded C language** using Keil software.

Features of 8051 Microcontroller:

- It having four register banks
- 64K bytes on-chip programmable memory (ROM)
- 128 bytes on-chip data memory (RAM)
- Address bus is 16-bit unidirectional
- Data bus is 8-bit bidirectional
- 128 user defined flags
- 16 bit timers
- 32 general purpose registers each of 8-bit
- 8051 microcontroller offers a number of special features such as ADC, UARTs, Op-amp, etc.

Architectre of 8051

Basic components present internally inside 8051 Microcontroller architecture are:

CPU (Central Processing Unit): CPU act as a mind of any processing machine. It synchronizes and manages all processes that are carried out in microcontroller. User has no power to control the functioning of CPU. It interprets the program stored in ROM and carries out from storage and then performs it projected duty. CPU manage the different types of registers available in 8051 microcontroller.

Interrupts: Interrupts is a sub-routine call that given by the microcontroller when some other program with high priority is request for acquiring the system buses the n interrupts occur in current running program.

Interrupts provide a method to postpone or delay the current process, performs a sub-routine task and then restart the standard program again.

Types of interrupt in 8051 Microcontroller:

Let's see the five sources of interrupts in 8051 Microcontroller:

- Timer 0 overflow interrupt - TF0
- Timer 1 overflow interrupt - TF1
- External hardware interrupt - INT0
- External hardware interrupt - INT1
- Serial communication interrupt - RI/TI

Memory: For operation Micro-controller required a program. This program guides the microcontroller to perform the specific tasks. This program installed in microcontroller required some on chip memory for the storage of the program.

Microcontroller also required memory for storage of data and operands for the short duration. In microcontroller 8051 there is code or program memory of 4 KB that is it has 4 KB ROM and it also comprise of data memory (RAM) of 128 bytes

Bus : Bus is a group of wires which uses as a communication canal or acts as means of data transfer. The different bus configuration includes 8, 16 or more cables. Therefore, a bus can bear 8 bits, 16 bits all together.

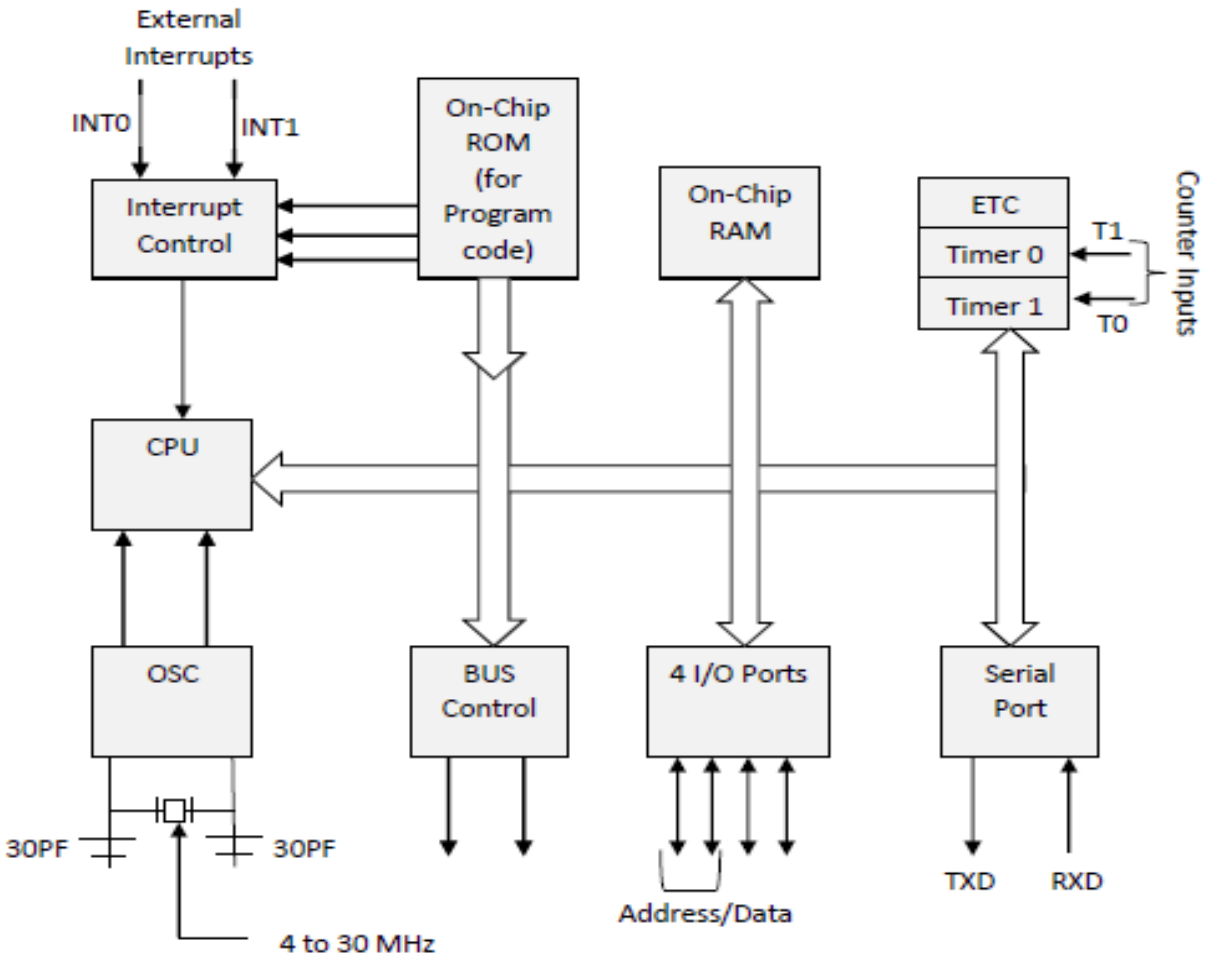
Types of buses in 8051 Microcontroller:

Let's see the two types of bus used in 8051 microcontroller:

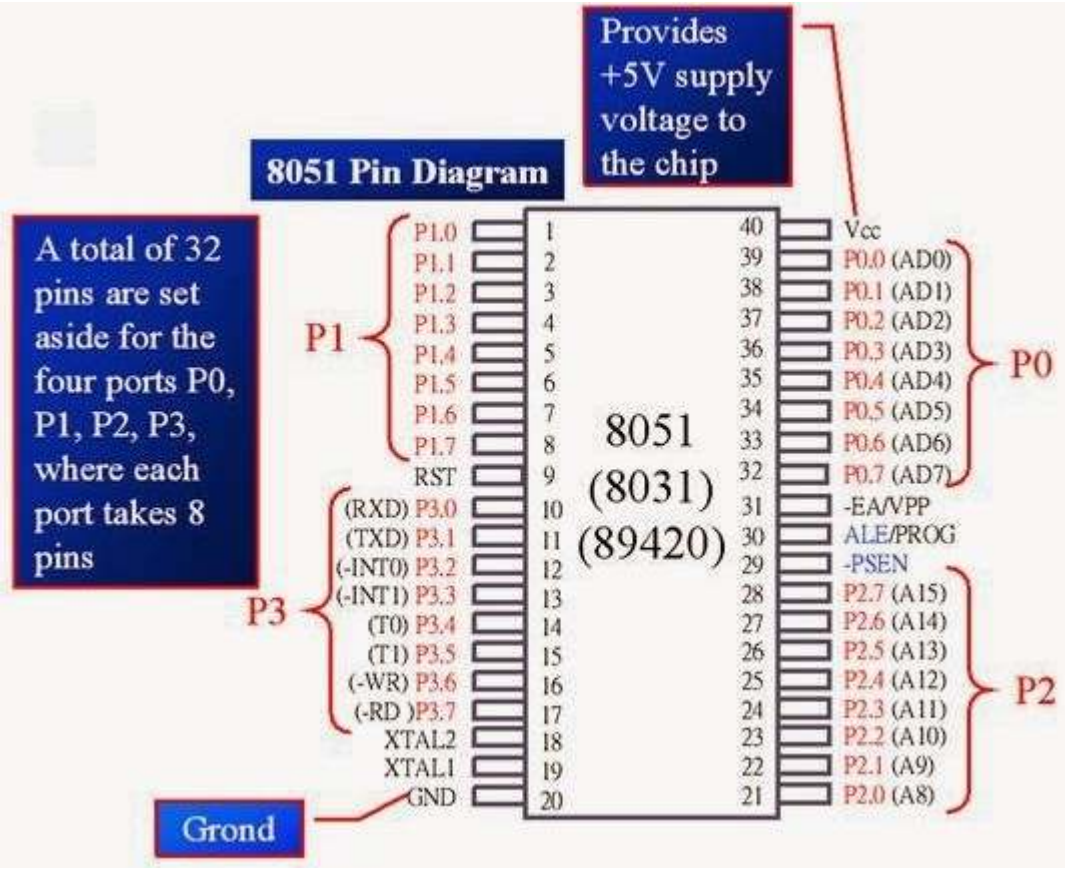
- **Address Bus**: 8051 microcontrollers is consisting of 16 bit address bus. It is generally be used for transferring the data from Central Processing Unit to Memory.
- **Data bus**: 8051 microcontroller is consisting of 8 bits data bus. It is generally be used for transferring the data from one peripherals position to other peripherals.

Oscillator: As the microcontroller is digital circuit therefore it needs timer for their operation. To perform timer operation inside microcontroller it required externally connected or on-chip oscillator.
Microcontroller is used inside an embedded system for managing the function of devices. Therefore, 8051 uses the two 16 bit counters and timers. For the operation of this timers and counters the oscillator is used inside microcontroller.

Architecure of 8051:



8051 Micro-controller Pin Description:



For explaining the pin diagram and pin configuration of microcontroller 8051, we are taking into deliberation a 40 pin Dual inline package (DIP). Now let's study through pin configuration in brief:-

Pins 1 – 8:- recognized as Port 1. Different from other ports, this port doesn't provide any other purpose. Port 1 is a domestically pulled up, quasi bi directional Input/output port.

Pin 9:- As made clear previously RESET pin is utilized to set the micro-controller 8051 to its primary values, whereas the micro-controller is functioning or at the early beginning of application. The RESET pin has to be set elevated for two machine rotations.

Pins 10 – 17:- recognized as Port 3. This port also supplies a number of other functions such as timer input, interrupts, serial communication indicators TxD & RxD, control indicators for outside memory interfacing WR & RD, etc. This is a domestic pull up port with quasi bi directional port within.

Pins 18 and 19:- These are employed for interfacing an outer crystal to give system clock.

Pin 20:- Titled as Vss – it symbolizes ground (0 V) association.

Pins- 21-28:- recognized as Port 2 (P 2.0 – P 2.7) – other than serving as Input/output port, senior order address bus indicators are multiplexed with this quasi bi directional port.

Pin- 29:- Program Store Enable or PSEN is employed to interpret sign from outer program memory.

Pin-30:- External Access or EA input is employed to permit or prohibit outer memory interfacing. If there is no outer memory need, this pin is dragged high by linking it to Vcc.

Pin-31:- Aka Address Latch Enable or ALE is brought into play to de-multiplex the address data indication of port 0 (for outer memory interfacing). Two ALE throbs are obtainable for every machine rotation.

Pins 32-39: recognized as Port 0 (P0.0 to P0.7) – other than serving as Input/output port, low order data & address bus signals are multiplexed with this port (to provide the use of outer memory interfacing). This pin is a bi directional Input/output port (the single one in microcontroller 8051) and outer pull up resistors are necessary to utilize this port as Input/output.

Pin-40: termed as Vcc is the chief power supply. By and large it is +5V DC.

Application of 8051 Microcontroller

The 8051 microcontroller applications include a large amount of machines because it is used for incorporating inside a project or to assemble a machine using it.

Let's see the major applications of 8051 Microcontroller:

1. **Energy Management** : In energy management system the measuring device is used for calculating the energy consumption in industrialized and domestic applications. These systems are manufactured by integrating the microcontrollers inside their architecture configuration.
2. **Automobiles** : Microcontroller 8051 is to be used for providing automobile solutions. They are largely be used in hybrid motor vehicles to control engine variations.
3. **Touch screens**: The advanced degree of microcontroller integrate the touch sensing ability within their design .Transportable devices such as cell phones, media players and gaming devices are some example of microcontroller integrated with touch screens.
4. **Medical Devices**: Microcontroller is used in various medical devices such as glucose and blood pressure measurement machine for monitoring and measuring the exact result in real-time computational environment.

QUESTIONS:

1. What is microprocessor?
2. What is microcontroller?
3. Basic use of 8051 microcontroller?